

**Debreceni Egyetem
Informatikai Kar**

DIGITÁLIS FOTÓK RETUSÁLÁSA

**Témavezető:
Dr. Hajdu András
Egyetemi adjunktus**

**Készítette:
Eszenyi László
Programozó
matematikus**

**Debrecen
2008**

Köszönöm Hajdu András témavezetőnek a szakdolgozatom elkészítésében nyújtott segítségét.
Továbbá köszönöm öcsémnek, Eszenyi Tamásnak a Szegedi Tudományegyetem fizikus
hallgatójának a segítségét.

Tartalomjegyzék

Tartalomjegyzék	3
Bevezetés.....	5
Fizikai háttér	6
Lencsetorzítás	6
Vignettálás	8
Perspektívorzítás	11
Színek	12
A vörös szem effektus	15
A korrekciók megvalósítása.....	16
Pont-transzformációk	16
A transzformációk alkalmazása előre illetve visszafele	16
Töredezettség	18
Supersampling	19
Lencsetorzítások korrekciója	20
A λ meghatározása	21
A korrekció véghezvitele adott λ esetén	23
A vignettálás korigálása	24
A perspektívorzítás korrekciója	25
A céltégalap előállítása	26
Sík centrális vetítése síkra	28
Négyszög leképezése négyszögre	29
Szín transzformációk	30
ColorMatrix	30
Szürkeskálás és szépia képek előállítása	31
Világosság	32
Kontraszt.....	32
Színtelítettség	33
A módosítások végrehajtása, összetett transzformációk	34
A vörös szem effektus korrekciója	35
Bőrszín detektálása	35
Viola – Jones objektumdetektálás	37
A szemek lokalizálása	39
Korrekció	41
Implementáció.....	42
A fejlesztői környezet	42
EasyPhotoRetouch.....	43
Képek betöltése, mentése	43
Szín transzformációk	43
Perspektívorzítás	46
Lencsetorzítások	48
Vágás	50

Vignettálás	52
Vörös szem	53
Összefoglalás	55
Irodalomjegyzék	56
Függelék.....	57
További tesztképek és eredmények	57

Bevezetés

Napjainkban a digitális fényképezés aranykorát éljük. Talán nincs is már olyan háztartás, ahol ne lenne legalább egy fényképező, ha nem is a régről ismert alakjában, de egy mobiltelefon formájában biztosan. Minek is köszönhető ez a népszerűség? A recept egyszerű: bárki számára könnyen elérhető, olcsó, használata nem igényel semmilyen fotográfiai előképzettséget, és ami talán a legfontosabb, hogy nem kell kétszer átgondolni a beállítást egy-egy látvány megörökítéséhez, nem egy huszonnégy vagy harminchat kockás filmre dolgozunk, hanem egy több száz kép tárolására alkalmas memóriára ahonnan a félre sikerült képek könnyen kitörölhetők. Ezen gépek túlnyomó része kompakt, objektívük beépített és általában képesek néhányszoros zoomra, mely a lencserendszer fókusz távolságának változtatásával lehetővé teszi a gépek minél szélesebb körű alkalmazását. A lencsék egyre kisebbek miközben látószögük egyre nagyobb lesz, ami biztosítja, hogy viszonylag kis távolságból is nagy teret képezhessünk le. Általában beépített vakut is tartalmaznak melyek adott esetben megfelelő fényt biztosítanak a felvételhez. Továbbá több-kevesebb automatika segíti a felhasználót a minél jobb eredmény eléréséhez.

Itt már gondolhatnánk azt is, hogy kis gépünk mindent tud, jobb eredményeket nem is lehet elérni, ez a fotózás. Igen ám, de akkor miért léteznek többszázezer forintos "nagygépek" cserélhető optikával, külső nagyteljesítményű vakuval, színszűrőkkel és egyéb tartozékokkal? A kisgépeknek van egy nagy hátránya, ami éppen az előnye is, a széleskörű alkalmazhatóság. Bármilyen téma, kompozíció megörökítésére alkalmas, de tulajdonképpen egyikre sem tökéletesen. Fényérzékenységük nem túl nagy, a beépített vaku csak két-három méteren belül hatékony, és ott sem mindig a kívánt hatást érjük el vele, illetve a nagy látószögű lencsék miatt torzulás jelentkezhet a képen ami adott esetben zavaró lehet.

Szakdolgozatom célja, hogy az olcsó, kompakt digitális fényképezők által készített képeken megjelenő leggyakoribb hibákat, illetve nem kívánt jelenségeket képfeldolgozási eszközök segítségével korrigáljam. A korrekciók végrehajtására egy könnyen kezelhető szoftvert készítettem.

A dolgozatomban a következő jelenségek bemutatására és korrekciójára térek ki:

- Lencsetorzítások – minden lencsére jellemző, a nagy látószögűekre fokozottan, a képen hordó vagy párna alakú torzulás jelentkezik
- Vignettálás – a lencse és az objektív más összetevőinek jellemzői miatt a kép a szélek felé haladva egyre sötétebb.
- Perspektívatorzítás – nem hiba, inkább jelenség mely adott esetben esztétikailag kifogásolható. Például egy magas épület közelről történő fotózásakor a képen az épület “hanyatt dől”.
- Színkorrekciók – az alacsony fényérzékenység miatt a kép sötét, a színek telítetlenek, a kontraszt alacsony.
- Vörös szem effektus megszüntetése - a beépített vakuk hátránya, hogy egyenesen a modell szemébe villannak ahonnan vörös fényben verődnek vissza az objektívbe.

Az algoritmusok implementálására Microsoft Visual C# -ot használtam.

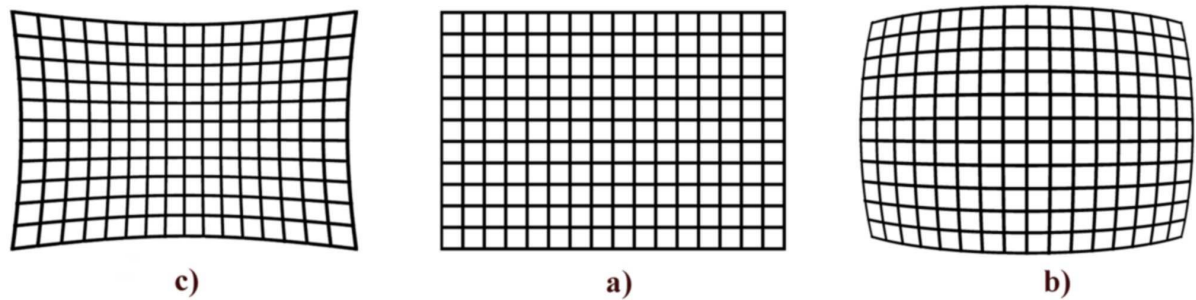
Fizikai háttér

Lencsetorzítás

Fényképezéskor a céltárgyáról visszaverődő fény (vagy esetlegesen az általuk kibocsátott fény) a fényképező objektívén keresztül az úgynevezett CCD érzékelőre jut, ahol megtörténik az adott kép digitalizálása, melynek során a folytonos kép egy diszkrét pontthalmazra képződik le, illetve a folytonos színtartomány szintén diszkrét intenzitásértékekre bomlik fel. Az objektív feladata tehát a nagy kiterjedésű, háromdimenziós téma leképezése a síkbeli CCD-re. Az objektív egy vagy több optikai lencséből, illetve tükrökből épülhet fel. A gyakorlat a lencse főtengelyétől távoli tárgyak éles leképezését és a kellő fényerő biztosítására nagy nyílású, széles sugárnyalábok alkalmazását követeli meg. Ekkor viszont a lencsék alapvető leképzési törvényszerűségeitől eltérések, leképzési hibák (képhibák) mutatkoznak, melyek nagyrészt a törőfelületek és a fénysugarak szimmetriatulajdonságaival összefüggő geometriai okokból származnak [1].

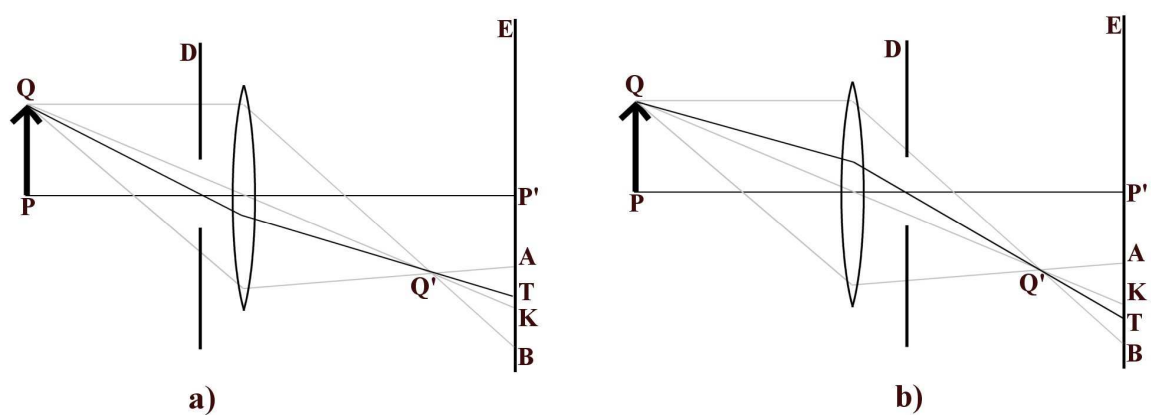
Az egyik ilyen képhiba a torzítás, mely a tárgynak a képhez való geometriai hasonlóságára vonatkozó hiba, amely akkor jön létre, ha az oldalnagyítás nem ugyanakkora a

kép minden részén. Megkülönböztetünk hordó ill. párna alakú torzítást aszerint, hogy az oldalnagyítás a széleken kisebb vagy nagyobb, mint középen. A következő ábrán látható egy négyzetes hálózat torzítás nélkül, hordó és párnatorzítással.



1.ábra: a) nincs torzítás; b) hordó alakú; c) párna alakú torzítás

Az objektívekben a lencséken kívül úgynevezett diafragmák is találhatók. Ezek kör alakú nyílások, melyek belső átmérője meghatározza, hogy mennyi fény haladhat át rajtuk. Ilyen diafragma a blende is, melynek belső átmérője változtatható, amit a fényképezők a fénymérést követően automatikusan határoznak meg. A diafragmákat általában a kép élességének javítására használják. Elhelyezkedhetnek a lencse előtt vagy után is, amely meghatározza a torzítás alakját. A következő ábrák szemléltetik, hogy ha a lencse elé tesszük a diafragmát, akkor hordó, ha mögé, akkor párna alakú torzítás jön létre.



2.ábra: a) hordó alakú torzítás; b) párna alakú torzítás

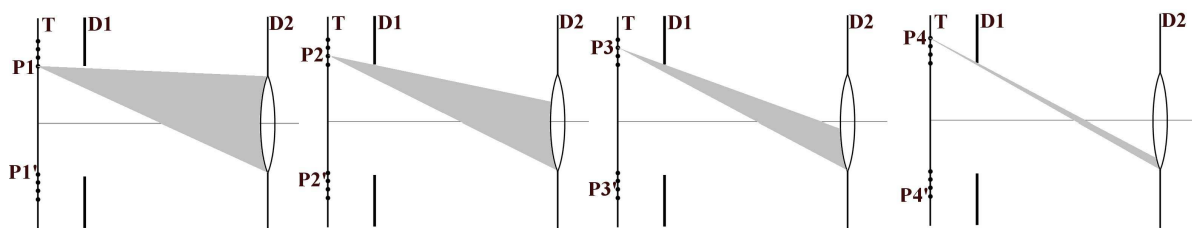
A 2. ábra bal oldala a hordó alakú torzítás kialakulását szemlélteti. Az ábrán a PQ tárgy leképezése történik meg az E ernyőre, ahol a D diafragma a tárgy és a lencse között helyezkedik el. A halványabb vonalak illusztrálják a leképezésnek azt az esetét, mikor nincs ott a D diafragma. Látható, hogy ebben az esetben a Q pont az AB fényfoltra képződik le, melynek optikai súlypontját a lencse közepén áthaladó sugárnak az E-vel való K metszéspontja jelöli ki. Ebben az esetben a kép torzításmentes lenne. A D diafragma alkalmazása miatt, az AB-nél kisebb fényfolt optikai súlypontja azonban a diafragmanyílás közepén áthaladó fősugárnak az E-vel alkotott T metszéspontja, amely a K-nál beljebb van, ez pedig a szélső képrészek összehúzódásának felel meg. A b) ábra a párnatorzítást szemlélteti, melynek magyarázata hasonló az előzőhöz. Ebben az esetben a K pont helyett a K-tól kintebbi T pontba képződik le a P, mely a széleken levő képrészek kinyúlását eredményezi.

A torzítás csökkentése vagy akár teljes megszüntetése speciális lencserendszerek alkalmazásával érhető el. Ilyen például az ortoszkópikus objektív, melyet többek között reprográfiai célokra lehet alkalmazni. Ez azonban a szokásos értelemben vett fotózásra nem alkalmas. Az optikai technika fejlesztői a leképzési hibák korrigálásában rendkívüli eredményeket értek el, de minden korrigált lencse általában csak annak a célnak felel meg teljesen, amelyre tervezték.

A feladat tehát adott: olyan eljárás kifejlesztése, mely a különböző mértékben, hordó vagy párna alakban torzult képekből előállítja a torzulásmentes képet.

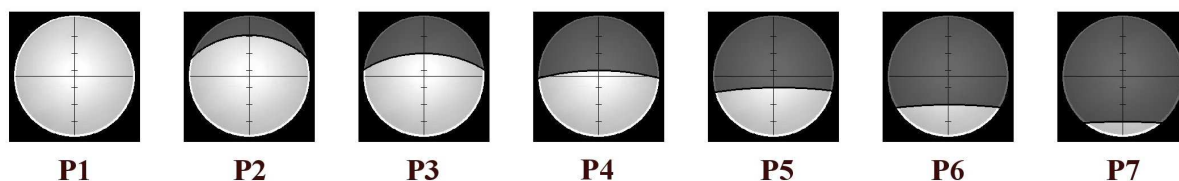
Vignettálás

A vignettálás egy gyakran előforduló jelenség, melynek meglétére azonban általában csak speciális esetekben figyelünk fel. A jelenség a következő: a kép szélei felé haladva, egy ponttól kezdve folytonosan és elég drasztikusan csökken az intenzitás, amely főleg a kép sarkaiban vehető észre. Ez változatos háttér esetében lehet, hogy fel sem tűnik a szemlélőnek, azonban, ha egy statikus háttér (például fal) előtti tárgyat fotóz, akkor igen zavaró lehet a besötétedő képsarkok látványa.



3.ábra: a lencsére jutó fénymennyiség

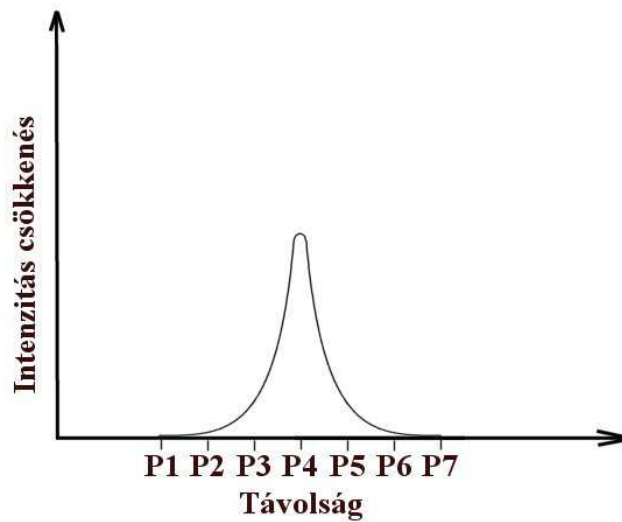
A fenti ábrán látható, hogy a P_1 és a P_1' -n kívül eső pontokból kiinduló sugárnyalábok egyre nagyobb részét takarja ki a D_1 diafragma, melynek következtében egyre kevesebb fény halad át a lencsén és így egyre csökkenő intenzitással képződnek le a pontok az ernyőre [1]. Látható az is, hogy míg a P_1 és a P_1' -től kifelé haladva csökken a pontok képének intenzitása, addig a P_1 és a P_1' közötti részre nincs hatással a diafragma. Az ábrából az is látszik, hogy a P_1 és a P_1' -től kifelé haladva a pontokból a lencsére jutó sugárnyalábok nyílása egyre kisebb, ennek következtében a lencsére jutó fénymennyiség csökken, azaz csökken a pontok képének intenzitása. Az ábra alapján azt várnánk, hogy az intenzitás az optikai tengelytől való távolság függvényében lineárisan változik. Ez az összefüggés a valós háromdimenziós esetben azonban nem ilyen egyszerű. A következő ábra ezt igyekszik szemléltetni. Kiinduló helyzetben a P_1 pontból a D_1 diafragmán át látjuk a lencsét, legyen ez az optikai tengelytől legtávolabbi olyan pont, melyből még látszik a teljes lencse (a diafragma nincs rá hatással). Ezután haladunk egyenletes lépésközökkel kifelé (távolodunk az optikai tengelytől), láthatóvá válik, hogy az egyes pontokból nézve a diafragma a lencse mekkora részét takarja ki.



4.ábra: A lencse látható része a különböző pontokból

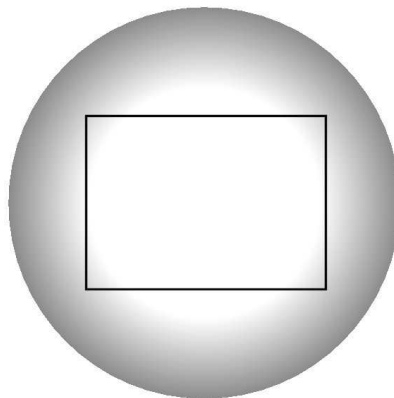
Az előző képsorozaton jól látszik, hogy bár egységnyi lépésközökkel távolodunk, lépésenként nem állandó a lencse látható felületének csökkenése. A kezdeti csökkenés kicsi, majd folyamatosan egyre nagyobb lesz, míg el nem érjük azt a pontot, ahonnan a lencse

felületének már csak fele látszik (P₄). Ettől az értéktől a fogyás egyre kisebb mértékű lesz, míg végül teljesen eltűnik a lencse. A következő grafikon az intenzitás csökkenés mértékét szemlélteti az optikai tengelytől való távolság függvényében, a jelzett pontok megfelelnek az előző ábra pontjainak.



5.ábra: az intenzitáscsökkenés mértéke a távolság függvényében

Az elkészült felvételeken természetesen nem figyelhető meg a teljes skála (intenzitáscsökkenés kezdetétől az eltűnésig), mivel a CCD-re a lencse által leképezett képnek csak egy része kerül, ahogy az a következő ábrán is látható. Így máris világos, hogy miért sötét a kép a sarkokban.



6.ábra: a kép sarkai besötétednek

Feladat: lehetővé tenni, hogy a kép középpontjától bizonyos távolságon kívül eső pontok intenzitása a távolságuk arányában növelhető legyen.

Perspektívorzítás

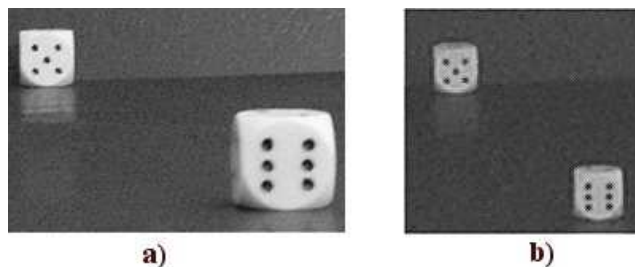
A perspektívorzítás a lencsetorzításokkal és a vignettálással szemben nem hiba, de adott esetben zavaró lehet. A perspektíva az emberek számára természetes jelenség, hiszen mi is így látjuk a világot. A lencsékhez hasonlóan a szemünk is úgy működik, hogy a távolabbi tárgyakat kisebbnek látja, mint a közelebbieket. Így fordulhat elő az is, hogy a távolba tartó párhuzamos egyenesek összetartanak, ez jól megfigyelhető például egy sínpár esetén.



7.ábra: egy sínpár perspektivikus képe

Az optikai rendszerek esetén is jelentkezik ez a hatás, melyet a lencsetorzítások és a vignettálás esetén is felelős diafragmák megléte és helyzete nagyban befolyásol [1]. Ha a diafragmát közvetlenül a lencse után tesszük, akkor a lencséhez közelebbi azonos méretű tárgy képe nagyobb lesz, mint a távolabbié. A diafragmát fokozatosan távolítva a lencsétől a két tárgy mérete közeledik egymáshoz, mikor pedig elérjük a lencse képoldali gyújtótávolságát a két tárgy mérete azonos lesz a képen is. Az utóbbi módon helyezik el a diafragmát a mérő mikroszkópokban, ahol fontos, hogy a tárgy távolságának megváltozása ne befolyásolja a képének nagyságát. A fényképezőkben is jelentkezik a perspektívorzítás, mellyel életszerű felvételeket produkálhatunk. Az objektív fókusztávolságának

változtatásával, változik a perspektívtorzítás mértéke. Ha ráközelítünk valamire, akkor a perspektíva mértéke sokkal kisebb, mintha ráközelítés nélkül fotózzuk ugyanazt a témát.



8.ábra: a) közelről zoom nélkül; b) távolról ráközelítve

A fényképezők perspektívtorzítása tehát öröndetes tény a valóságú ábrázolás érdekében. Előfordulnak azonban olyan helyzetek, melyekben nem szeretnénk hatását érzékelni. Ilyen például, mikor egy magas épületet fotózunk, viszonylag közelről és alulról, mert egyszerűen más beállítás nem lehetséges az adott körülmények között. Természetesen ezt a képet kicsi fókusztávolsággal fogjuk készíteni (nem közelítünk rá), hogy az épület minél nagyobb része látható legyen a felvételen. Ebben az esetben viszont a tőlünk távol levő teteje az épületnek annyira összemegy a képen, hogy egészen azt a hatás kelti, mintha hanyatt akarna dőlni.

A cél tehát az, hogy a perspektívtorzulást szenvedett képet oly módon alakítsuk át, hogy az eredménykép torzulásmentes legyen. Ezzel az eszközzel természetesen nem kell minden esetben élni, hiszen azzal esetleg csökkenthetjük a kép valószerűségét, de adott esetben jól jöhet.

Színek

Fényképeink esztétikai értékét nagyban befolyásolják színi tulajdonságai. Ezek többek között a következők: világosság, színtelítettség, kontraszt és színárnyalat.

A világosság a vizuális érzékelés egyik jellemzője, amely a fényforrás által kibocsátott fény mennyiséget fejezi ki. Esetünkben a fényforrások az egyes képpontok, melyek

világossága - ha az RGB színskálát tekintjük - a három színikoordináta következő kombinációja [2]: $\mu = \frac{R + G + B}{3}$

Amatőr fotózáskor gyakran előforduló probléma, hogy nem megfelelő a fény mennyisége, ezért a képek túl világosak vagy éppen túl sötétek lesznek. Ez köszönhető a gyengébb minőségű fényképezőknek vagy egyszerűen a téma rossz beállításának. Például fénnel szemben fotózunk, vagy túl sötét van és a kis teljesítményű vaku nem biztosít elegendő fényt. A feladat igen egyszerű, szükség van egy eljárásra, mellyel a nem megfelelő intenzitású képek világosságát emelni vagy csökkenteni lehet. Ezzel könnyen korrigálható a kép egészére jellemző világossági probléma.

Az alacsony színtelítettség az egyik legáltalánosabb problémája az egyszerű fényképezőgépekkel készült képeknek. A színek tompák, fakók, amit lehet, hogy észre sem veszünk mindaddig, míg egy megfelelő szoftverrel nem emeljük utólagosan a kép színtelítettségét. Ekkor derül ki igazán, hogy mennyi életet lehet belevinni egy átlagos képbe, és hogy milyen színtelen is volt az eredeti képünk. A színtelítettség definíció szerint azt fejezi ki, hogy a különböző színekben a domináns színárnyalat mennyire dominál a többi színárnyalathoz képest.

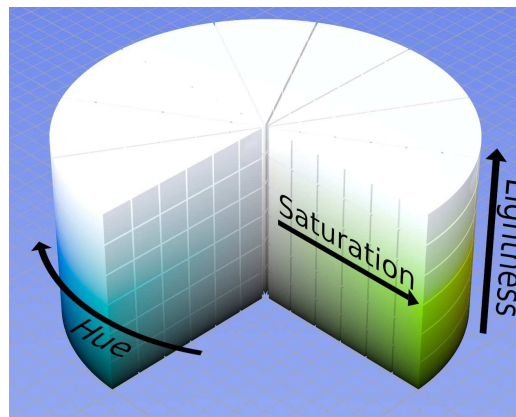
A színárnyalat azt fejezi ki, hogy az egyes színekben mely hullámhosszok dominálnak. A színárnyalat határozza meg, hogy egy adott szín vörös, kék vagy például sárga. A különböző színárnyalatok spektruma a szokásos módon így jeleníthető meg:



9.ábra: a színárnyalat spektruma

A színárnyalat, színtelítettség, világosság tulajdonságok önmaguk is meghatározzák az adott pontok színét. A HSL (Hue, Saturation, Lightness) színtér [3] pontosan ezeken a paramétereken alapul. Természetesen minden RGB színtérben ábrázolt koordinátahármashoz egyértelműen meghatározható a neki megfelelő HSL térbeli koordinátahármas és fordítva. A fentebb vázolt tulajdonságokat a HSL térben nagyon egyszerűen lehet változtatni. Ha például szeretnénk megnövelni a színtelítettséget, akkor nincs más dolgunk, minthogy megnöveljük

az adott pont vagy pontok S koordinátáját. A probléma csupán azzal van, hogy a pixelek általában RGB színtérben vannak ábrázolva. Ahhoz, hogy ebben a környezetben, az előzőekben leírt egyszerű módon megváltoztathassuk valamely paramétert, először az adott pont RGB koordinátáit át kell váltani HSL-beli koordinátákra, ott el lehet végezni a műveletet, majd ismét vissza kell konvertálni RGB koordinátákra, hogy ábrázolni tudjuk. Mindez elég költséges műveletsorozat, hatékonyabb módot kell találni, melynek kulcsa, hogy az RGB színtérben maradunk.



10.ábra: a HSL színtér

A kontraszt alatt a kép világos és sötét részei közti különbséget értjük. Ha ugyanarról a tárgyról szólva az egyik képen nagyobbak a tónusok különbségei, akkor beszélünk nagyobb kontrasztról. A közepes tónusok eközben lehetnek változatlanok, míg a kontrasztosabb képeken elvesznek a legvilágosabb és legsötétebb részletek, árnyalatok. A csekély kontrasztú felvételek lágyak, erőtleneek, szürkék, mintha életlenek lennének. A kontrasztot többféle módon definiálják, néhányan meghatározónak tartják a színeket, mások csak a világosság értékekkel számolnak. Egyik legismertebb definíciója a Michelson-kontraszt [4], melynek képlete a következő: $\frac{L_{\max} - L_{\min}}{L_{\max} + L_{\min}}$, ahol L_{\max} a maximális világosságérték, míg L_{\min} a minimális világosságérték. Fotóink feljavításának érdekében általában emelni kell a kontrasztot, mely megvalósítható úgy, hogy egy bizonyos értéknél sötétebb pontokat még sötétebbé teszünk, míg a korlátnál világosabbakat még világosabbá.

Célom, hogy a három alapvető problémára, a világosság, kontraszt, színtelítettség nem megfelelő értékeinek módosítására, egy egyszerű lehetőséget biztosítsak a felhasználók számára.

A vörös szem effektus

A vörös szem effektus kompakt, beépített vakuval rendelkező kisépek esetén gyakori problémát jelent. Ez a jelenség a vaku által a szembe jutó fény visszaverődéséből alakul ki, melynek vörös színe a retina mögötti véredényeknek köszönhető. A kialakulásában alapvetően három fő tényező játszik szerepet: a szem felépítése, a fényképező felépítése és a beállítási körülmények. Kialakulásának megakadályozása érdekében e tényezők megismerése szükséges.

A szem felépítése adott, természetesen nem lehet változtatni rajta. Vakut általában akkor használunk, ha a kép készítéséhez különben nincs elegendő fény. Ilyen sötét környezetben a pupilla kitágul, hogy több fény juthasson a szembe, ez a tény viszont elősegíti az effektus kialakulását. Továbbá gyakrabban jelentkezik a probléma világos szemű emberek és kisgyerekek esetén. A világosabb színű szivárványhártyában kevesebb a színt meghatározó melanin, így több fény jut a szembe. A gyerekek pupillája pedig jobban ki tud tágulni, mint a felnőtteké. Néhány tudományos alkalmazásban szándékosan idézik elő a vörös szem effektust, ilyen a szembetegségek felderítése és a kormeghatározás.

A beépített vakuval rendelkező fényképezők hátránya, hogy nagyon közel van a vaku az objektívhez, mely elősegíti, hogy a szemekből visszaverődő fény egyenesen a gépbe jusson. Célszerű - ha van - bekapcsolni a vörös szem effektust csökkentő opciót, ami a felvétel előtt néhány elővillanás vagy folyamatos világítás segítségével összehúzódnásra készíti a pupillát, így kevesebb fény jut majd a retinára és csökken a vörös fényvisszaverődés. Az opció hátránya, hogy a fotózás alanyának néhány másodpercig pózolni kell, ami például kisgyerekek esetén nem biztos, hogy lehetséges.

Az effektus a megfelelő beállítási körülményekkel nagyban csökkenthető. Minél világosabb van annál szűkebbek a pupillák, tehát ha lehetséges fel kell kapcsolni minden lámpát. További lehetőség, hogy ne nézzünk a fényképezőbe, akkor nem oda fog

visszaverődni a vaku fénye, nézzük inkább a fotós vállát, vagy a fényképező vakuval ellentétes oldalát.

A vörös szem effektus kialakulásának megakadályozására a fentiekben vázolt feltételek megléte szükséges, mely elég nagy előkészületet igényelhet. Ez azonban nem mindig lehetséges, és ekkor nem marad más lehetőség, mint az utólagos javítás. Céлом, hogy egy teljesen automatikus eljárást fejlesszek ki, mellyel a felhasználó egy gombnyomásra ki tudja javítani képeit, ezzel időt takaríthat meg és nem szükséges, hogy jártas legyen a retusálásban.

A korrekciók megvalósítása

Pont-transzformációk

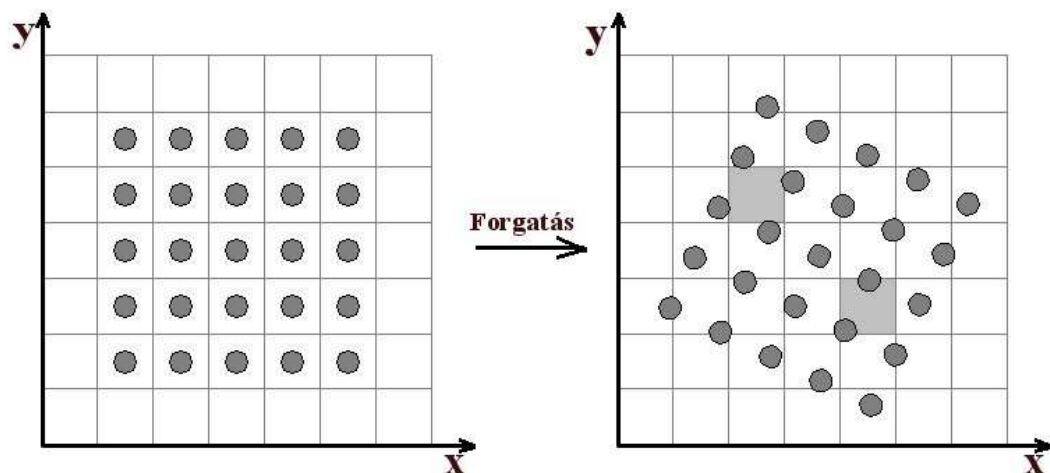
A lencsetorzítások és a perspektívorzítás korrekciója során pont-transzformációk alkalmazása szükséges. Ez azt jelenti, hogy a kép pontjaihoz egy transzformációs függvény segítségével új pontokat rendelünk hozzá. Ilyen transzformációk például a számítógépes grafikában alkalmazott eltolás, skálázás, forgatás stb. A transzformációt egy függvény definiálja, melyet a grafikus objektum összes pontjára egyesével kell alkalmazni. Ezeket a transzformációkat széles körben alkalmazzák poligonokra, ahol a függvényértéket csak a poligonok csúcsaira kell meghatározni, majd a megfelelő csúcsok éllel összeköthetők. Ebben az esetben azonban a kép nem poligonokból, hanem pontokból épül fel, tehát egy adott transzformációt a kép összes pontjára alkalmazni kell. Ezzel olyan problémák merülnek fel, melyek a poligonok csúcsainak transzformációja során nem.

A transzformációk alkalmazása előre illetve visszafele

Az angol szakirodalomból az előre illetve a visszafele irányuló transzformációkat forward mapping és reverse mapping néven ismerjük. A forward mapping esetén a

transzformációs függvényt a kiindulási kép minden pontjára alkalmazzuk, és ezzel megkapjuk a célképet. A reverse mapping esetén viszont a függvény inverzét alkalmazzuk a célkép minden pontjára, annak érdekében, hogy minden pontnak megkeressük a kiinduló képbeli pozícióját. Természetesen ez a módszer csak invertálható függvények esetén hajtható végre.

Miért van kétféle megközelítés? A kérdést úgy is feltehetnénk, hogy mi szükség van a reverse mapping használatára, mikor a forward mapping használata kézenfekvő? A válasz a raszteres megjelenítés tulajdonságaiban keresendő. A valóságban folytonos objektumok képe egy diszkrét ponthalmazra van leképezve, ami általában egy négyzetrács. A négyzetrácsot felépítő négyzetek a képpontok, melyek koordinátája a négyzetrácsban való elhelyezkedésüket (sor, oszlop) jelöli. A transzformációk alkalmazása során általában semmi sem garantálja, hogy az egyes pontok képe pontosan a négyzetrács egy négyzetére esik, tehát azt, hogy egész koordinátákat kapunk eredményül.



11.ábra: forgatás után a szürkével jelölt négyzetekbe vagy több pont esik vagy egy sem

A 11. ábrán egy egyszerű forgatás eredményét látjuk. Látható, hogy míg a kiindulási képen minden négyzetben egy pont volt, addig az eredmény képen van olyan négyzet melyben több pont is van, illetve van olyan amiben egy sincs. Ez azt jelenti, hogy a transzformáció alkalmazása során, a kép raszteres reprezentációjának következtében egyes

pontok képe ugyanaz lett, illetve az eredményképről pontok hiányoznak. Mindez a forward mapping-nek köszönhető. A probléma megoldása a reverse mapping használata.

A reverse mapping során a célkép pontjait bejárva, az inverz transzformáció segítségével minden ponthoz megkeressük a neki megfelelő kiinduló képbeli ősképet. Ekkor az előzőekben vázolt problémák megszűnnek. A négyzetrács minden négyzetében csak egy pont lesz, illetve az objektumból nem lesznek hiányzó pontok.

Ezért használom én is a reverse mapping-et. Ennek egyetlen nagy hátránya, hogy a transzformációt leíró függvényt invertálni kell, ami adott esetben elég bonyolult függvényt ad eredményül, ezzel megnövelve a transzformációs eljárás költségét.

Töredezettség

A valóságban folytonos vonalak a raszteres megjelenítésnek köszönhetően nem ábrázolhatóak folytonosan, csak töredezetten (aliasing). Ennek ellenére a töredezett vonal melletti pontok színének megfelelő beállításával a hatás csökkenthető, illetve a szem számára észrevehetetlenné tehető (antialiasing).

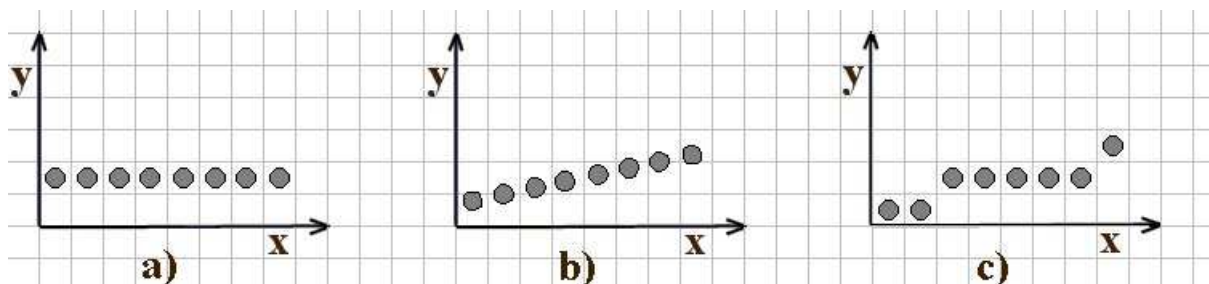


12.ábra: párhuzamos egyenesek, az alsón látható az antialias hatása

A valós, folytonos képek digitalizálása után ez a hatás nem érzékelhető, azonban, ha alkalmazunk a képre valamilyen transzformációt, a töredezettség problémája azonnal felmerül.

A probléma oka, hogy az egész koordinátás pontok transzformációja után általában a pont koordinátái nem lesznek egészek. Ezek viszont nem ábrázolhatóak, ezért valamilyen eljárással helyettesíteni kell őket. Legegyszerűbb eljárás a kerekítés, melynek során a valós koordinátákat egészsre kerekítjük, és ez már ábrázolható. Ebben az esetben máris bekövetkezik

a fenti ábrának megfelelő aliasing hatás. A következő ábrán látható, hogy egy egyenes forgatása után mely valós pontokat kapjuk eredményül, illetve, hogy azokat mely egész koordinátás pontokra kerekítjük.



13.ábra: a) az eredeti szakasz; b) a szakasz képe forgatás után valós koordinátákkal; c) a szakasz képe forgatás után egészre kerekített koordinátákkal

Ennek kiküszöbölése érdekében különböző technikákat használnak. A legelterjedtebbek a következők: bilineáris interpoláció, bikubikus interpoláció, supersampling.

Supersampling

Az általam is használt supersampling [7] módszer lényege, hogy a kép felbontását megnöveljük, elvégezzük rajta az adott transzformációt, majd a felbontást visszaállítjuk az eredetire, melynek során kiküszöböljük a töredezettséget.

A supersampling menete a gyakorlatban, reverse mapping esetén:

Végigjárjuk a célkép pontjait az eredeti felbontásnak megfelelően. Minden egyes pontot felosztunk valahány részre (például: 3x3), az így kapott valós koordinátás pontokra elvégezzük az inverz transzformációt, majd a kiinduló képen az egészre kerekített koordinátáknak megfelelő pontok színének átlagát véve, megkapjuk az adott célképbeli pont színét.

A módszer előnye, hogy egyszerűen kivitelezhető. Hátránya a meglehetősen nagy számításigény. Egy 800x600-as kép esetén, ha minden pontot 3x3-as egyenlő részre osztunk fel, a kép felbontása 2400x1800-ra változik, a pontonkénti transzformáció műveletigénye 9-szeresére nő az átlagolástól eltekintve.

Lencsetorzítások korrekciója

A hordó és a párna alakú torzításokat nem véletlenül emlegetik egyszerre, hiszen nem csak kialakulásuk fizikai háttere közel azonos, hanem a leggyakrabban használt matematikai modelljük [5] is:

$$u = x(1 + \lambda_1 r^2 + \lambda_2 r^4 + \dots), \quad v = y(1 + \lambda_1 r^2 + \lambda_2 r^4 + \dots) \quad (1)$$

Ahol az (x,y) torzult pont torzulásmentes megfelelője az (u,v) pont. A $\lambda_1, \lambda_2, \dots$ a torzítás paraméterei. Az r pedig az (x,y) pont távolsága a torzítás centrumától, mely szokásos módon a kép középpontja.

$$r^2 = x^2 + y^2 \quad (2)$$

Természetesen a kép középpontjában van a (0,0) pont, ebben a koordináarendszerben értendők az (x,y), (u,v) pontok.

Az (1) képletet általában legfeljebb a negyedrendű taggal bezárólag közelítik, vagy az alap modell helyett az osztási modellt használják, mely kis torzítás esetén hasonlóan jó eredményeket szolgáltat, viszont a számítási igényt jelentősen lecsökkenti.

$$u = \frac{x}{1 + \beta r^2}, \quad v = \frac{y}{1 + \beta r^2} \quad (3)$$

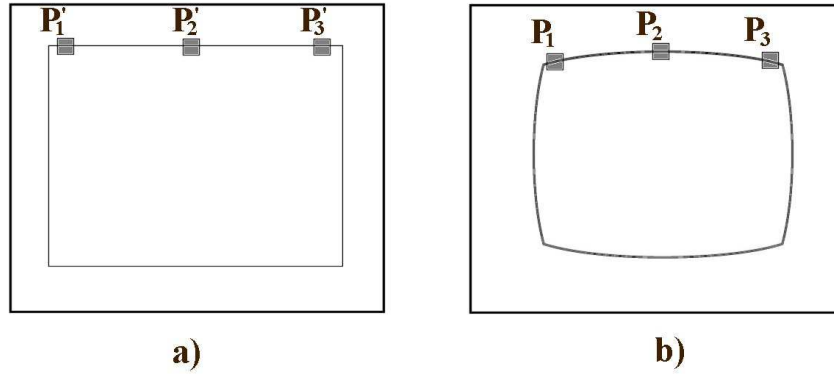
Ez a módszer azonban nagy torzítás esetén nagyon pontatlan eredményt ad. Éppen ezért én az eredeti képlet közelítését használom a másodrendű taggal bezárólag.

$$u = x(1 + \lambda r^2), \quad v = y(1 + \lambda r^2) \quad (4)$$

Ez a képlet már meglehetősen jó eredményeket szolgáltat (nagy torzítás esetén is). A képletből kiderül, hogy pozitív λ esetén hordó-, negatív λ esetén párnatorzításról beszélünk. Ha a felhasználó megad egy λ értéket, akkor elvégezhető a torzítás, a kép korrigálható. Céлом azonban az is, hogy a felhasználó ne csak a λ -át adhassa meg paraméterként, hanem egy torzult egyenest is, mely alapján automatikusan határozza meg az eljárás a megfelelő λ -át.

A λ meghatározása

Az eljárásomban λ meghatározására a felhasználónak nincs más dolga, mint kijelölni a képen három pontot, melyeknek a torzulásmentes képen egy egyenesbe kellene esniük. A következő ábra ezt szemlélteti.



14.ábra: a) egy téglalapot ábrázoló torzulásmentes kép; b) ugyanaz a téglalap hordótorzítással

A cél az, hogy olyan λ -át találjunk, amivel a (4) képlet alapján egy egyenesre illeszkedő pontokat kapunk. A kijelölt pontok a következők: $p_1=(x_1,y_1)$, $p_2=(x_2,y_2)$, $p_3=(x_3,y_3)$. A nekik megfelelő torzítatlan pontok homogén koordináták alakban, (4) alapján:

$$p'_1 = \begin{pmatrix} x_1 \\ y_1 \\ 1 \\ \frac{1}{1 + \lambda r_1^2} \end{pmatrix}, \quad p'_2 = \begin{pmatrix} x_2 \\ y_2 \\ 1 \\ \frac{1}{1 + \lambda r_2^2} \end{pmatrix}, \quad p'_3 = \begin{pmatrix} x_3 \\ y_3 \\ 1 \\ \frac{1}{1 + \lambda r_3^2} \end{pmatrix} \quad (5)$$

Egy pont rajta van egy egyenesen, ha a pont homogén koordináták vektorának és az egyenes

homogén koordináták vektorának skaláris szorzata nulla. Legyen $L = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}$ egyenes, ekkor

célunk szerint teljesülnie kell az alábbi egyenlőségeknek:

$$p_1^T * L = 0, \quad p_2^T * L = 0, \quad p_3^T * L = 0 \quad (6)$$

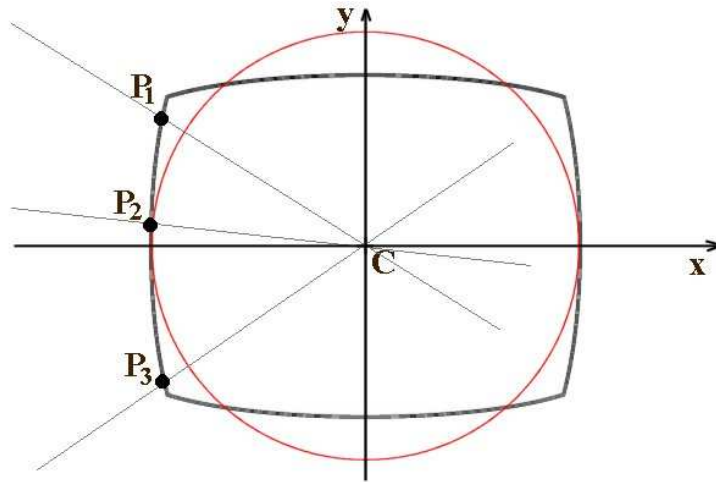
A p_1', p_2', p_3' pontokat egy mátrixba rendezve a (6) egyenletrendszer a következőképpen is felírható:

$$\begin{pmatrix} x_1 & y_1 & \frac{1}{\lambda r_1^2} \\ x_2 & y_2 & \frac{1}{\lambda r_2^2} \\ x_3 & y_3 & \frac{1}{\lambda r_3^2} \end{pmatrix} * \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (7)$$

Ez az egyenlőség csak akkor teljesülhet, ha a mátrix determinánsa nulla, mivel sem a mátrix nem nullmátrix, sem az L nem nullvektor. Tehát a következő egyenletnek kell teljesülnie:

$$\frac{1}{1+\lambda r_1^2}(x_2 y_3 - y_2 x_3) - \frac{1}{1+\lambda r_2^2}(x_1 y_3 - y_1 x_3) + \frac{1}{1+\lambda r_3^2}(x_1 y_2 - y_1 x_2) = 0 \quad (8)$$

A (8) egyenletből kifejezhető λ . Ekkor λ -ra másodfokú egyenletet kapunk, melynek megoldásai λ_1 és λ_2 . A (4) egyenletbe bármelyik λ -át helyettesítve, a kapott pontok egy egyenesbe esnek. Azonban a mi esetünkben csak az egyik szolgáltat megfelelő eredményt.



15.ábra

A megfelelő λ kiválasztásának szemléltetésére legyen adott a 15. ábrán látható eset. A p_1, p_2, p_3 pontokat kell egy egyenesbe hozni. A (4) képletekből következik, hogy a pontok a szürkével ábrázolt egyeneseken mozgathatók, amiket a pontok és a C torzítási középpont határoz meg. A piros színnel ábrázolt kör azt szemlélteti, hogy a p_1, p_3 szélső pontok távolabb helyezkednek el a torzítás középpontjától, mint a p_2 . Ekkor viszont (4)-ből következik, hogy tetszőleges λ -ra p_1, p_3 “gyorsabban” mozog az egyenesek mentén, mint a közbülső p_2 pont. Ha

a C irányába mozgatjuk a pontokat (negatív λ), akkor a p_1 , p_3 pontok gyorsabban közelednek C-hez, aminek következtében a p_1 , p_2 , p_3 által meghatározott görbe még görbébb lesz. Egy egyenesbe csak akkor kerülhetnek, ha valamely szélső pont(ok) átlépik C-t, viszont az így kapott egyenes a mi esetünkben nem megfelelő. Ha a C-től távolítjuk a pontokat (pozitív λ), akkor p_1 , p_3 gyorsabb mozgása miatt lesz egy olyan λ , melyre a három pont egy egyenesbe esik, ez éppen az az egyenes, amit keresünk. Akár távolítjuk a pontokat C-től, akár közelítjük őket, mindkét irányban csak egy-egy λ lesz, melyre a három pont egy egyenesbe kerül, ugyanis miután elérték ezt az egyenest, a szélső pontok gyorsabb mozgása miatt soha többé nem alakulhat ki olyan helyzet, melyben egy egyenesre illeszkednek.

Az előző magyarázat után tehát a λ_1 illetve a λ_2 értékek közül azt kell választani, melynek hatására (4) alapján egyik pont sem lépi át a C torzítási középpontot.

A korrekció véghezvitele adott λ esetén

Ha megvan a λ torzítási paraméter - akár explicit módon megadva, akár az előző részben taglalt módon kiszámítva – beírható a (4) képletbe, és a transzformáció elvégezhető a torzult kép minden (x,y) pontjára. Azonban, ha a (4) képletet alkalmazzuk az eredeti képre, akkor a Pont-transzformációk című fejezetben leírtak miatt az eredmény képen nem kívánt hatások lépnek fel. Ezért szükség van a (4) képlet invertálására, majd az inverz függvény célképre történő alkalmazására.

Most mutatkozik meg a (4) képlet egyetlen hátránya a (3)-al szemben, ugyanis a (4) egyenletrendszer x -re illetve y -ra történő megoldása során harmadfokú egyenletre jutunk, melynek megoldása sok műveletet igényel.

Az egyenletrendszer megoldása során a harmadfokú egyenlet u -tól, v -tól függetlenül mindig a következő alakú lesz:

$$A^3 + pA + q = 0 \tag{9}$$

Látható, hogy a másodrendű tag hiányzik. A harmadfokú egyenlet megoldásával az egyes esetekben x -re vagy y -ra kapunk eredményt, melyből a másik már könnyen kifejezhető.

Az egyes esetek a következők:

i) Ha $u, v \neq 0$, akkor a harmadfokú egyenlet a következő:

$$x^3 + px + q = 0, \text{ ahol } p = \frac{vu^2}{\lambda vu^2 + \lambda v^3} \text{ és } q = -\frac{vu^3}{\lambda vu^2 + \lambda v^3}.$$

x meghatározása után, $y = \frac{vx}{u}$ kiszámítható.

ii) Ha $u = 0$ és $v \neq 0$, akkor $x = 0$, y pedig a következő egyenlet megoldásából áll elő:

$$y^3 + py + q = 0, \text{ ahol } p = \frac{1}{\lambda} \text{ és } q = -\frac{v}{\lambda}$$

iii) Ha $u \neq 0$ és $v = 0$, akkor $y = 0$, x pedig a következőképpen áll elő:

$$x^3 + px + q = 0, \text{ ahol } p = \frac{1}{\lambda}, q = -\frac{u}{\lambda}$$

iv) Ha $u, v = 0$, akkor $x, y = 0$, a torzítás centrumában vagyunk.

A vignettálás korrigálása

A korrekció a vignettálási effektus tulajdonságai miatt lokális világosságmanipulációt igényel. Mivel az effektus csak a kép sarkaiban, esetlegesen szélein jelentkezik, így csak az ott levő pontok világosságértékét kell módosítani. Ez a módosítás a világosságértékek növelését jelenti, mégpedig a pont pozíciójának függvényében.

A vignettálás tulajdonságai miatt, a kép centrumától – ami az optikai tengely dőféspontja a CCD-n – egy bizonyos távolságon belül elhelyezkedő pontok világosságértékeire nincs hatással az effektus. Ezen pontokkal tehát nincs dolgunk, változatlanul hagyjuk őket. Az adott távolságon kívül eső pontok, melyeket a korrekció érinteni fog a következők:

$$\left\{ P \mid \sqrt{(P_x - C_x)^2 + (P_y - C_y)^2} > T \right\}, \text{ ahol } C \text{ a kép középpontja, } T \text{ pedig egy távolság.} \quad (1)$$

A világosságértékek csökkenése a C-től T távolságra kezdődik el, és folyamatosan tart egészen a kép széléig. Annak biztosítására, hogy a középponttól legtávolabbi pontokban is (a kép sarokpontjaiban) megfelelő legyen a világosságérték, adjunk meg egy B értéket, mely az az érték, amivel ezen pontok világossága fog növekedni. Legyen a sarokpontok távolsága C-től mT.

Már tudjuk, hogy a C-től T távolságra levő pontok világosságát még 0-val, míg az mT távolságra levőket B-vel fogjuk növelni. A feladat már csak annyi, hogy meghatározzuk a

világosságnövelés mértékét a T és mT távolságok között elhelyezkedő pontokra. A legegyszerűbb megoldás az lenne, ha egyenletesen növelnénk 0-ról B -re a világosságnövekedés mértékét a pontok C -től való távolsága alapján (lineáris interpoláció). Ez azonban – mint ahogyan a vignettálás fizikai hátterét leíró részben is kiderül – nem ad korrekt eredményt, szabad szemmel is jól látszik a nem megfelelő hatás. A vignettálás című fejezetben ábrázolt grafikon alapján sokkal jobb eredmény érhető el, ha négyzetesen közelítünk a következő szerint:

$$I = \frac{(aktT - T)^2}{(mT - T)^2} B, \text{ ahol } aktT \text{ az aktuális pont távolsága } C\text{-től. Az } I \text{ nem lesz más mit az adott}$$

pontban a világosságérték növekménye, tehát a pont világosságát ezzel kell növelni. Látható, ha $aktT = T$, akkor $I = 0$, ha $aktT = mT$, akkor pedig $I = B$ -t kapunk, a közbűlső pontokban pedig négyzetesen nő a világosítás mértéke.

A perspektívectorzítás korrekciója

A perspektívectorzítás korrigálása során azt szeretnénk elérni, hogy a képen látható, vagy esetleg képzeletbeli egyenesek egymással bezárt szögét megváltoztassuk. Például, ha egy épület homlokzatát lefotózzuk, akkor a képen az épület ablakainak párhuzamos szélei összetarthatnak, ezzel a valóságban egymásra merőleges szélek közbezárt szöge megváltozik.



16.ábra: *egy ablak perspektivikus képe*

Az előző képen látható, hogy az ablak szemközti szélei nem párhuzamosak, a szomszédos szélek pedig nem derékszöget zárnak be. Hozzá kellene rendelni valahogy az ablak sarkait a képen látható téglalap sarkaihoz, és ennek megfelelően úgy torzítani a képet, hogy a sarkok a megfelelő helyre kerüljenek. Adott tehát a feladat: egy tetszőleges négyszöget (ablak) képezzünk le egy téglalapra, vagy általánosabb esetben egy másik tetszőleges négyszögre. A következőkben ennek megvalósítását olvashatjuk.

Először is a kiindulási és a cél négyszöget meg kell adni valahogyan. Erre több lehetőség is kínálkozik. A legkézenfekvőbb megoldás az, hogy legyen az eredeti négyszög maga a kép, tehát annak négy sarka által meghatározott téglalap. Ezután mondjuk meg, hogy melyik sarok hova kerüljön, ezzel definiálva a cél négyszöget. A cél négyszög explicit megadásával a felhasználó “manuálisan” határozza meg a transzformációt, például egérrel megfogja a kép valamelyik sarkát és kihúzza.

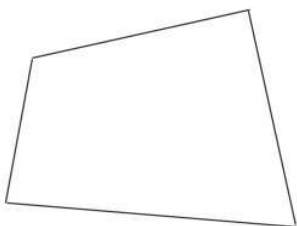
Egy másik lehetséges módszer, hogy a felhasználó kijelöl egy, a képen látható négyszöget, melyről azt szeretné, hogy téglalap legyen. Például, egérekattintással kijelöli az ablak sarkait. Ekkor az algoritmus feladata, hogy “automatikusan” előállítson egy megfelelő téglalapot, melyre megtörténhet a négyszög áttranszformálása.

Mindkét eljárást implementáltam a megbízhatóság (manuális) és a kényelem (automatikus) jegyében.

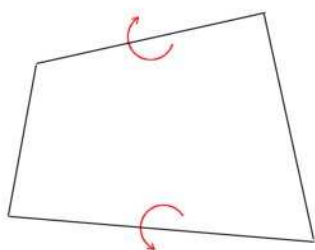
A céltéglalap előállítása

Adott a felhasználó által kijelölt tetszőleges négyszög, ehhez kell egy megfelelő téglalapot találni, amire majd a leképezés megtörténhet. A probléma azzal van, hogy a négyszög alapján nem tudunk eleget a fotós, tárgyhoz viszonyított pozíciójáról, illetve a tárgy eredeti méretarányairól. Éppen ezért egy, az eredetinek pontosan megfelelő oldalarányú téglalap megadása nem lehetséges. Ennek tudatában a következő eljárás csak egy “lehetséges” megoldást szolgáltat:

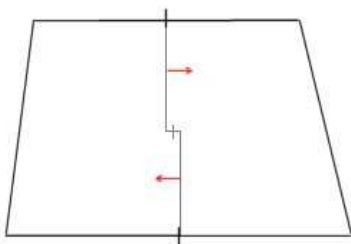
i) Adott a következő négyszög:



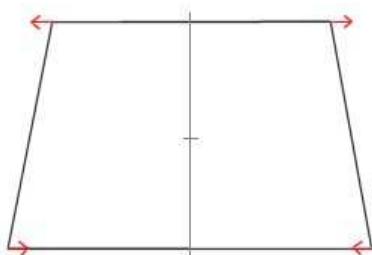
ii) Az alsó és felső éleket a saját középpontjuk körül forgassuk vízszintesbe



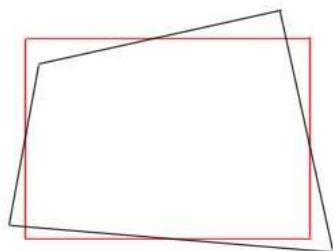
iii) Az alsó és felső élek középpontját toljuk közös függőleges egyenesbe



iv) Az alsó és felső élek hossza legyen a hosszuk átlaga, miközben a középpontjuk pozíciója változatlan



Az eljárás végén létrejövő téglalapra fogjuk leképezni a kijelölt négyszöget.



Sík centrális vetítése síkra

Sík képeken a szokásos háromdimenziós megjelenítés centrális vetítéssel (projekcióval) érhető el, melynek során a világ-koordinátarendszerbeli háromdimenziós pontokat egy transzformációval a kép kétdimenziós koordinátarendszerébe képezzük. A leképezés homogén koordináták alakban a következőképpen írható fel:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = P_{3 \times 4} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}, \quad (1)$$

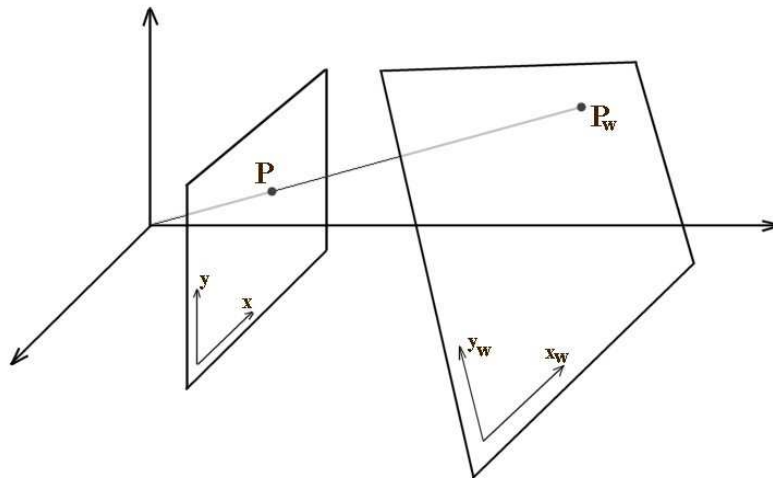
ahol P a centrális projekció mátrixa, (x_w, y_w, z_w) a világ-koordinátarendszerbeli pont, (x, y) pedig a neki megfeleltetett síkbeli képpont.

Legyen $z_w = 0$ minden világ-koordinátarendszerbeli pontra, ekkor tulajdonképpen egy síkot kapunk, melynek centrális projekciója a képsíkra ugyanúgy elvégezhető, mint az (1)-ben, de egyszerűsíthető is $z_w = 0$ miatt. Ekkor ezt kapjuk [6]:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H_{3 \times 3} \begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix} \quad (2)$$

$$H_{3 \times 3} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}, \text{ ahol } h_{33} = 1 \quad (3)$$

Ezt az esetet a következőképpen ábrázolhatjuk:



17.ábra: centrális vetítés

A 17. ábrán látható p_w és p pontok közötti összefüggés a következő:

$$x = \frac{h_{11}x_w + h_{12}y_w + h_{13}}{h_{31}x_w + h_{32}y_w + h_{33}}, \quad y = \frac{h_{21}x_w + h_{22}y_w + h_{23}}{h_{31}x_w + h_{32}y_w + h_{33}} \quad (4)$$

A szokásos esetben p_w és H adott, melyekből meghatározható p . Tegyük fel, hogy most nem ismerjük H -t, viszont ismerjük p_w -t és képét p -t. Látható, hogy adott p_w, p pontpár esetén a (4) alapján két egyenlet írható fel, a H mátrixban azonban 8 ismeretlen van, melynek meghatározására 8 egyenletre van szükség. Ebből következik, hogy négy pontpárra van szükség a meghatározására. Továbbá a négy-négy pontnak általános helyzetűnek kell lennie, tehát semelyik három nem eshet közülük egy egyenesre.

Tehát adott négy pontpár, ebből nyolc egyenlet, aminek az egyik lehetséges megoldási módja a következő:

$$\begin{pmatrix} x_{w1} & y_{w1} & 1 & 0 & 0 & 0 & -x_1x_{w1} & -x_1y_{w1} \\ 0 & 0 & 0 & x_{w1} & y_{w1} & 1 & -y_1x_{w1} & -y_1y_{w1} \\ x_{w2} & y_{w2} & 1 & 0 & 0 & 0 & -x_2x_{w2} & -x_2y_{w2} \\ 0 & 0 & 0 & x_{w2} & y_{w2} & 1 & -y_2x_{w2} & -y_2y_{w2} \\ x_{w3} & y_{w3} & 1 & 0 & 0 & 0 & -x_3x_{w3} & -x_3y_{w3} \\ 0 & 0 & 0 & x_{w3} & y_{w3} & 1 & -y_3x_{w3} & -y_3y_{w3} \\ x_{w4} & y_{w4} & 1 & 0 & 0 & 0 & -x_4x_{w4} & -x_4y_{w4} \\ 0 & 0 & 0 & x_{w4} & y_{w4} & 1 & -y_4x_{w4} & -y_4y_{w4} \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{pmatrix} \quad (5)$$

Négyszög leképezése négyszögre

Mi is volt a feladat? Manuális esetben a kép téglalapjának leképezése egy, a felhasználó által megadott tetszőleges négyszögre. Automatikus esetben pedig a felhasználó által kijelölt tetszőleges négyszög leképezése téglalapra. Mindkét esetben valamilyen négyszög leképezése

egy másikra, tehát négy tetszőleges, általános helyzetű pont (sarkok) leképezése, négy másik tetszőleges, általános helyzetű pontra. A feladat a (3)-ban megadott mátrix meghatározása, mely az (5) alapján kivitelezhető. Ha megvan a H mátrix, akkor a kép minden pontját beszorozva vele, megkapjuk a kívánt eredményképet. Természetesen itt is inverz transzformációt érdemes alkalmazni a célkép pontjaira – a Pont-transzformációk fejezetben leírtak miatt –, ami a célkép pontjainak H^{-1} -el való szorzását jelenti.

Szín transzformációk

A GDI+ API többféle pixelformátumot ismer. Tekintsük most az általam is használt Format24bppRgb formátumot. Ebben a formátumban egy pont 24 biten van ábrázolva, nyolc-nyolc biten a vörös, kék és zöld komponensek. Ebből következik, hogy egy-egy komponens értéke 0 és 255 közötti értékeket vehet fel. Ezen értékek együttese határozza meg az adott pontok színét. Amennyiben meg akarjuk változtatni egy pont színét, az előbbi komponensek értékét kell változtatni a megadott tartományon belül. Ha egy kép egészére vonatkozóan szeretnénk színmanipulációt végezni, akkor a kép minden egyes pontjának egyenként kell megváltoztatni a színét.

ColorMatrix

A ColorMatrix osztály [9] segítségével 5x5-ös mátrixokat hozhatunk létre, melyekkel az ImageAttributes osztály az adott képen színmanipulációkat tud végrehajtani. A kép összes pontjára elvégzi a transzformációt a mátrix alapján. A következő műveletet hajtja végre minden pontra:

$$(R' \ G' \ B' \ A' \ f) = (R \ G \ B \ A \ 1) \cdot CM \quad (1)$$

Ahol $(R \ G \ B \ A \ 1)$ a pont eredeti színkomponenseinek homogén koordinátás sorvektora, az A nem más, mint az alfa komponens értéke, de ez 24 bites formátum esetén nem játszik szerepet. CM a mátrix, $(R' \ G' \ B' \ A' \ f)$ pedig a pont komponenseinek új értéke, szintén homogén koordinátás alakban. A művelet elvégzése előtt az RGBA értékeket 0 és 1 közé normalizálja, majd a számítás végeztével visszaalakítja 0 és 255 közé.

Cél az, hogy adott manipulációhoz megkonstruáljuk a megfelelő ColorMatrix objektumot, mellyel azután egyszerűen és gyorsan elvégezhető a transzformáció a kép összes pontjára.

Szürkeskálás és szépia képek előállítása

Színes képekből szürkeskálás képet a szokásos eljárás szerint a következő képlettel lehet előállítani:

$$I = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B \quad (2)$$

Ahol RGB a pont eredeti színtkomponenseinek értéke, I pedig a szürkeskálás pont intenzitása lesz. Ha a szürkeskálás képpontokat 8 biten ábrázoljuk, akkor az adott pont egyetlen komponense I lesz. Ha pedig az eredeti 24 bites képen szeretnénk ábrázolni a szürkeárnyaltos pontot, mint ebben az esetben is, akkor a pont RGB komponensei mindegyikének az I értéket kell adni.

Ebből következően a ColorMatrix így néz ki:

$$CM_{grey} = \begin{pmatrix} 0.3 & 0.3 & 0.3 & 0 & 0 \\ 0.59 & 0.59 & 0.59 & 0 & 0 \\ 0.11 & 0.11 & 0.11 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

A szépia képek régi időket idéző barna árnyaltos színeit, a szürkeárnyaltos képekhez hasonló módon lehet előállítani, azonban itt az RGB komponensek értékei különbözőek lesznek, annak érdekében, hogy a barnás színeket megjeleníthessék.

$$CM_{sepia} = \begin{pmatrix} 0.393 & 0.349 & 0.272 & 0 & 0 \\ 0.769 & 0.686 & 0.534 & 0 & 0 \\ 0.189 & 0.168 & 0.131 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

Világosság

Egy színes pont világosságértékét számíthatjuk akár a (2) képlet alapján is, de akár egyenlő súlyokkal is vehetjük az egyes komponenseket, ekkor a következő képletet kapjuk:

$$I' = \frac{R + G + B}{3} \quad (5)$$

Utóbbi esetben, ha növelni, vagy csökkenteni szeretnénk a pont világosságát, megtehetjük, hogy minden komponenst ugyanazzal az értékkel (b) növelünk, vagy éppen csökkentünk.

Ekkor a pont komponenseinek új értékei a következők:

$$\begin{aligned} R' &= R + b \\ G' &= G + b \\ B' &= B + b \end{aligned} \quad (6)$$

Ennek megfelelően egyszerűen felírható a ColorMatrix:

$$CM_b = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ b & b & b & 0 & 1 \end{pmatrix} \quad (7)$$

Kontraszt

A kontrasztnak több definíciója is van, egyesek csak a világosságértékeket tartják meghatározónak, mások a színeket is. Én most az utóbbit választom, tehát a kontraszt módosításával az egyes pontok színe is változik.

Szürkeskálás képeken már könnyebb egységes definíciót adni a kontrasztra, ugyanis ott csak a pontok intenzitása lehet meghatározó. Ennek megfelelően kontrasztemelés céljából a sötétebb pontok intenzitását csökkenteni kell, míg a világosabb pontokét emelni. A következő néhány lépés ezt szemlélteti:

$$\begin{aligned} I' &= I / 255 \\ I'' &= I' - 0.5 \\ I''' &= I'' \cdot c \\ I'' &= I''' + 0.5 \\ I' &= I'' \cdot 255 \\ (c > 0) \end{aligned} \quad (8)$$

A (8) műveletsorozat végeztével I' lesz a pont új intenzitás értéke. Amennyiben c kisebb mint 1, akkor csökken a kontraszt, ha nagyobb mint 1, akkor növekszik. Látható, hogy a sötét és világos pontok közötti korlát a 0-255 tartomány közepe, tehát, ha egy pont intenzitása ettől nagyobb, akkor $c > 1$ esetén növekszik, míg $c < 1$ esetén csökken az intenzitása. Ha egy pont intenzitása a középvértéktől kisebb, akkor $c > 1$ esetén csökken, $c < 1$ esetén növekszik az intenzitása.

A szürkeskálás képen végezhető kontrasztmódosítás kiterjeszthető színes képekre is, úgy, hogy a (8)-ban leírt műveletsorozatot minden komponensre elvégezzük. Ennek alapján a mátrix a következő lesz:

$$t = 0.5 \cdot (1 - c),$$

$$CM_c = \begin{pmatrix} c & 0 & 0 & 0 & 0 \\ 0 & c & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ t & t & t & 0 & 1 \end{pmatrix} \quad (9)$$

Látható, hogy ugyanaz a műveletsorozat adódik minden komponensre, mint (8)-ban. A 255-el való osztást/szorzást automatikusan elvégzi.

Színeltéttség

A színeltéttséget a HSL skála használata nélkül is elég jó közelítéssel lehet változtatni. Egy adott pontunk színeltéttsége legyen S_I , értékét nem határozzuk meg. Tudjuk (2) alapján, hogy a pontnak megfelelő szürkeskálás értéke I , viszont színeltéttsége nulla. Következésképpen, ha a pont komponenseinek értékét közelítjük az I -hez, akkor csökkentjük a színeltéttséget, ha távolítjuk, akkor növeljük. Ez a gondolatmenet lineáris interpolációval illetve extrapolációval véghez vihető [8].

A következő példában a felére csökkentem egy pont színeltéttségét. Legyenek egy P pont komponenseinek értékei a következők: R,G,B, ekkor (2) alapján meghatározható I . Ezekből lineáris interpolációval a következő adódik:

$$s = 0.5$$

$$R' = (1-s) \cdot I + s \cdot R$$

$$G' = (1-s) \cdot I + s \cdot G$$

$$B' = (1-s) \cdot I + s \cdot B$$

(10)

Ez a közelítés elég jó eredményeket ad, ráadásul kevesebb művelet végrehajtásával.

A ColorMatrix már nem olyan egyszerű, mint az előzőekben, de azért még kivitelezhető.

$$CM_s = \begin{pmatrix} a & b & c & 0 & 0 \\ d & e & f & 0 & 0 \\ g & h & i & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, ahol$$

$$a = (1-s)rw + s$$

$$b = (1-s)rw$$

$$c = (1-s)rw$$

$$d = (1-s)gw$$

$$e = (1-s)gw + s$$

$$f = (1-s)gw$$

$$g = (1-s)bw$$

$$h = (1-s)bw$$

$$i = (1-s)bw + s,$$

továbbá

$$rw = 0.3$$

$$gw = 0.59$$

$$bw = 0.11$$

(11)

Látható, hogy rw , gw , bw a (2)-ből ismert súlyok, az I meghatározásához kellenek.

A CM_s -sel való szorzás után R' -re a következő eredményt kapjuk:

$$R' = (1-s)(rw \cdot R + gw \cdot G + bw \cdot B) + s \cdot R.$$

A módosítások végrehajtása, összetett transzformációk

A felhasználó egy-egy előre megadott skáláról választhatja ki a világosság, kontraszt, színtelítettség paramétereit. A kiválasztott paramétertől függően előáll a megfelelő ColorMatrix, melynek segítségével elvégezhető a transzformáció. A transzformációt mindig

az eredeti képre végezzük, például ha megnöveljük a világosságot, akkor ennek megfelelően előáll egy eredménykép, ha viszont még tovább növeljük, akkor nem az eredményképre alkalmazzuk a transzformációt, hanem az eredetire a megnövelt értékkel.

Többféle módosítás is végrehajtható egy időben, például módosítjuk a világosságot, majd a kontrasztot. Ekkor a következő művelet hajtódik végre a kép pontjaira:

$$((P_R \ P_G \ P_B \ P_A \ 1) \cdot CM_b) \cdot CM_c \quad (12)$$

A (12) a mátrixok szorzásának tulajdonságai miatt a következő módon is felírható:

$$(P_R \ P_G \ P_B \ P_A \ 1) \cdot (CM_b \cdot CM_c) \quad (13)$$

Tehát, ha többféle módosítást is szeretnénk végrehajtani, akkor az egyes módosításoknak megfelelően előállítjuk a mátrixokat, ezeket összeszorozzuk, majd az így kapott mátrix segítségével egy lépésben hajtjuk végre az összes módosítást, ezzel elkerülve azt, hogy a képet pontról-pontra többször is végig kelljen járni.

A vörös szem effektus korrekciója

Az eljárás két fő lépésre osztható: először meg kell találni a vörös szemeket, majd jöhet a korrekciós fázis. A szemek detektálása két lépésben történik, először arcokat keresünk, azután azokon belül lokalizáljuk a szemeket. A keresési és korrekciós fázisok megvalósítására több lehetőség kínálkozik, a következőkben az általam kipróbáltakat vázolom.

Bőrszín detektálása

Az emberi bőr színe szerencsére elég jól definiálható, ezért egy átlagos képen elég nagy bizonyossággal megállapítható egy pontról, hogy lehet-e bőrpont vagy sem. Éppen ezért sok módszer létezik, melyek különböző metrikákat adnak arra vonatkozóan, hogy egy pont mennyire bőrszínű. Én egy egészen egyszerű módszert használtam, melynek előnye, hogy gyors, jó eredményeket ad egészen sötét bőrszíntől a világosig, hátránya – mint általában a színinformáción alapuló kereséseknek – hogy a detektálás során sok nem bőrpontot is bőrpontként azonosít. A módszer gyorsasága abból fakad, hogy explicit módon definiálja, hogy egy pont színkomponenseinek milyen feltételeknek kell megfelelni. Egy (R,G,B) színkomponensekkel rendelkező pont bőrpont, ha a következők teljesülnek [10]:

$$\begin{aligned}
 R &> 95 \\
 G &> 40 \\
 B &> 20 \\
 R &> G \\
 R &> B \\
 R - G &> 15 \\
 R - B &> 15
 \end{aligned}
 \tag{1}$$

Miután egy kép pontjait bőrszínű és nem bőrszínű osztályokba soroltuk, az összefüggő bőrszínű területeket tekinthetjük arcoknak, melyeken jelentkezhethet a vörös szem effektus. Természetesen lehet, hogy egy ilyen összefüggő terület nem arc, ekkor legfeljebb nem találunk majd rajta vörös szemet. Továbbá problémát jelenthet az is, hogy esetleg nem csak az arcot találjuk meg, hanem a hozzá tartozó testet is, vagy az arc összeolvad a háttérrel, melyek a továbbiakban csökkentik a szemek megtalálásának esélyét. Ugyanis, ha feltételezhetnénk, hogy minden esetben csak egy-egy arcot detektálunk, akkor előre megadhatnánk bizonyos információkat a szemekre vonatkozóan (pl. két szem távolsága), melyek segíthetnének a szemek pozíciójának felderítésében.

Tehát, ha a cél az arcdetektálás, akkor érdemes a bőrszín detektálást valamilyen más módszerrel kombinálni.



a)



b)

18.ábra: a) az eredeti kép; b) a bőrszínű pontok

Viola – Jones objektumdetektálás

Az eredetileg Viola és Jones által kifejlesztett [11], majd Leinhart által továbbfejlesztett algoritmust főként arcdetektálásra használják, de jól alkalmazható egyéb objektumok keresésére is. Az algoritmus [12] több osztályozó egymás utáni alkalmazásával hamar ki tudja zárni azokat a képrészleteket, melyek nem a keresett objektumot tartalmazzák, illetve nagy megbízhatósággal találja meg a keresett objektumot. Az eljárás szűrkeskálás képen dolgozik, de nem közvetlenül a pontok intenzitásával, hanem képrégiók jellemzőivel, melyek a Haar-féle bázisfüggvényekre hasonlítanak (Haar-like features). Meghatározásuk gyors, és használatukkal jobban jellemezhető egy objektum, mint csupán pontok segítségével. Az algoritmus a megfelelő jellemzők kiválasztására és az osztályozó betanítására az Adaboost tanuló algoritmust használja.

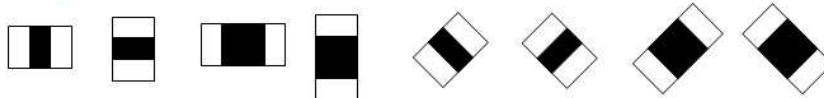
Haar-like features

A jellemzők két vagy három téglalap alakú tartományba eső pontok intenzitásával határozhatók meg. Az alábbi ábrán láthatók az egyes jellemzők.

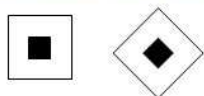
Él jellemzők



Vonal jellemzők



Centrum-környezet jellemzők

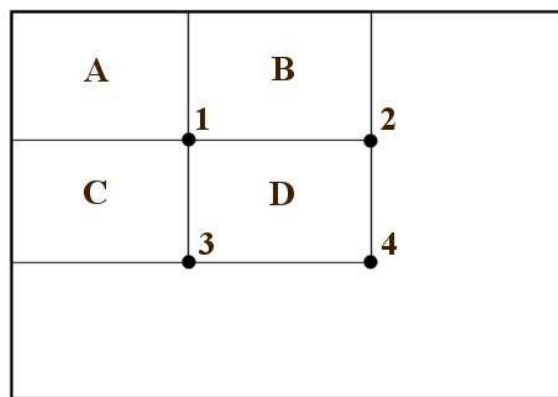


19.ábra: Leinhart által kiterjesztett jellemzők

Az egyes téglalapok által lefedett tartományban határozzuk meg a pontok intenzitásának összegét, majd a különböző színnel jelölt tartományokban kapott értékeknek vegyük a különbségét, az így kapott érték lesz a képrészlet adott jellemzőre vonatkozó értéke. Az egyes jellemzőkre vonatkozó érték a képinegrálnak köszönhetően gyorsan meghatározható. A képinegrált egyszer kell kiszámítani a feldolgozás elején, képlete a következő:

$$SAT(x, y) = \sum_{i \leq x, j \leq y} I(i, j) \quad (1)$$

A SAT (Summed Area Table) a számítógépes grafikában használt összegzett terület tábla, elve megegyezik az integrálképpel, ezért használatos az integrálkép jelölésére is. Az $I(i, j)$ nem más, mint az (i, j) pont intenzitása. A függvényérték tehát az (x, y) pontban egyenlő a (x, y) ponttól balra-fent elhelyezkedő pontok intenzitásának összegével. Ha megvan az integrálkép, az egyes téglalap alakú tartományokban néhány lépéssel meghatározható a pontok intenzitásösszege.



20.ábra: a D tartományban levő intenzitásösszeg meghatározása

Az ábrán látható, hogy a D tartományban levő pontok intenzitásának összege, három elemi művelettel meghatározható: $4+1-(2+3)$

Leinhart bevezette a 45 fokkal elforgatott képintegrált, melynek számítása hasonlóan egyszerű az előzőekhez.

Az Adaboost és a kiértékelés

A jellemzők megadása után megfelelő számú pozitív (arc) illetve negatív (nem arc) képet megvizsgálva az Adaboost algoritmus képes az osztályozó függvény előállítására. Az Adaboost [13] (Adaptive Boosting) “fokozott tanulás” lényege, hogy az első fázisban alkalmazott úgynevezett gyenge osztályozó minden jellemzőhöz hozzárendel egy optimális küszöbosztályozó függvényt, mely azonban még elég alacsony találati arányt tesz lehetővé. Második fázisként jön a felfokozó (boosting) eljárás, mely megkeresi a legmeghatározóbb jellemzőket, ezzel növelve a pozitív találati arányt és csökkentve a hibás találatok számát.

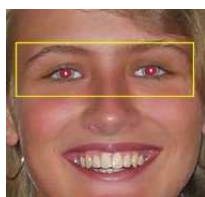
A megfelelő jellemzők kiválogatása után felépít egy egyszerű döntési fát, melyben az egyes kiértékelések során vagy megáll és elutasítja a bementként kapott képet, vagy tovább folytatja a kiértékelést egy újabb jellemző alapján.

A képek feldolgozása, az eljárás hátrányai

A cél az, hogy egy adott képen arcokat keressünk. Ennek érdekében egy kereső ablakot kell többféle méretben végigjáratni a képen és eldönteni, hogy az ablakban látható képrészlet lehet-e arc vagy sem. A méretezés viszonylag könnyen megvalósítható, tekintettel arra, hogy ebben az esetben nem a képet méretezzük át, hanem a jellemzőket meghatározó téglalapok méretét. Viszont abban az esetben, mikor az arc állása nem a betanításnak megfelelő, az algoritmus nem fogja felismerni. Ilyen eset lehet, ha a szemből nézett függőleges állású arcok alapján van betanítva, a képen pedig oldalra van billentve a fej, vagy nem szembe néz. Ennek megoldására utóbbiakat különböző objektumoknak kell venni, amire szintén be kell tanítani a detektálót, illetve az elforgatott fej esetén a képet többféle szögben el kell forgatni, és úgy is végre kell hajtani a detektálást, mindez jelentősen megnövelheti a feldolgozás idejét. A probléma csökkentésére megoldást nyújthat a bőrszín detektálással való kombináció, mellyel a keresést a bőrszínű területekre korlátozzuk.

A szemek lokalizálása

A szemek megkeresése viszonylag egyszerű feladat, ha feltételezzük, hogy az arcot már megtaláltuk. A szemek színe, alakja jól megkülönbözteti őket az arc egyéb részeitől. Munkám során a Viola – Jones algoritmust használtam frontális arcok megtalálására, mellyel egyéb, plusz információkat is nyertem a szemekre vonatkozólag. Egy szemből nézeti arcról tudjuk, hogy a szemeknek nagyjából hol kell elhelyezkedni, így a keresést egy kisebb régióra lehet szűkíteni. Az alábbi képen látható az a téglalap alakú tartomány, melyen a továbbiakban dolgozunk.



21.ábra: a szemeket tartalmazó terület

A problémából adódó tény, nevezetesen, hogy vörös szemeket kell keresni, plusz információkat szolgáltat a színre vonatkozóan, melyet szintén ki lehet használni. Az arc méreteiből meg lehet határozni, hogy a szemeknek nagyjából mekkoráknak kell lenni illetve, hogy milyen távolságban vannak egymástól. Továbbá tudjuk, hogy a pupilla kerek, így a keresett objektumok alakjáról is van egy jól meghatározott információ.

A keresést a színre vonatkozó információ alapján kezdem. Készítek egy vörösséget jellemző térképet, melyben az eredeti kép minden pontjára kiszámolom annak vörösségét a következő képlet alapján [14]:

$$redness(x, y) = \frac{R(x, y)^2}{G(x, y)^2 + B(x, y)^2 + K} \quad (2)$$

ahol K egy konstans, mellyel elkerülhető a nullával való osztás. A következő képen látható a kapott eredmény.



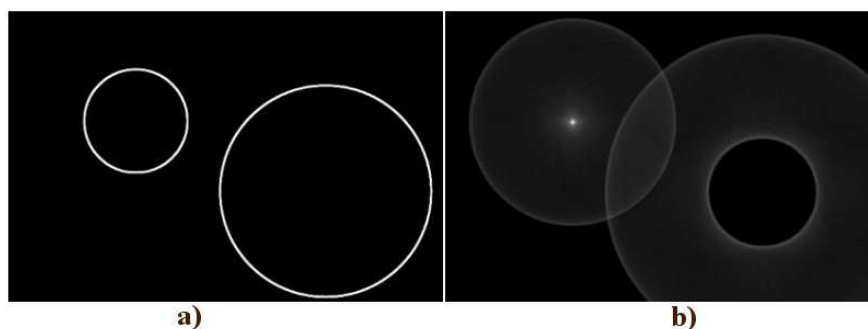
22.ábra: vörösség térkép

A következő lépésben azt használom ki, hogy kör alakú objektumokat keresek. A (2)-ben meghatározott vörösség térképen éldetektálást végzek, majd a kapott eredményen Hough kör transzformációt.



23.ábra: a vörösség térképről készült élkép

A transzformációval meghatározott r sugarú körök kereshetők. Az algoritmus során bejárjuk a képet, ahol nem nulla a pont intenzitása (vagy bináris képek esetén 1), ott megnézzük, hogy az adott pont mely r sugarú körökre esik, az ezen körök középpontját reprezentáló pontok intenzitását egyel növeljük. Ha a képen található r sugarú kör, akkor a Hough-transzformáltján a középpontjában egy magas intenzitásértékű pont lesz.



24.ábra: a) egy élkép; b) az élkép Hough-transzformáltja a kisebb kör sugarával

A pupilla sugarát azonban nem ismerjük, csak becsülni tudjuk, így az előbbi transzformációt több r értékre is elvégzem egy bizonyos tartományon belül. A transzformációval előálló körök közül azokkal dolgozom tovább, melyek intenzitása egy bizonyos értéknél nagyobb, ezek lesznek a lehetséges pupillák.

Előfordulhat még az éldetektálás során, hogy nem csak a pupillák körvonalát kapom meg, mivel gyakran elég nagy a szemöldökök körüli és a szemek szélein levő rész vörössége is. Ennek következtében a lehetséges pupillák között lehetnek olyanok, melyek ilyen élek miatt keletkeztek. Ezek elég jól kiszűrhetőek, azzal, hogy az általuk határolt terület vörössége nem elég nagy, vagy a közvetlen környezetükben hasonló vörösségértékek mutatkoznak. Továbbá, ha még mindig több mint két lehetséges pupillám van, akkor a nagyobb intenzitásúakat kell választani, össze lehet hasonlítani a méretüket, távolságukat, melyeket nagyjából behatároltam az arc méretéből.

Korrekción

A korrekciós lépés a kereséshez viszonyítva sokkal egyszerűbben megoldható. Tudjuk, hogy alap esetben a pupillának feketének kell lenni, így egyszerűen a keresésben meghatározott kör alakú tartományokat kifesthetjük feketére. Ez abban az esetben, ha kicsi a kép felbontása (vagy a kép egészéhez képest kicsi az arc), elfogadható eredményt ad, ám nagyobb felbontás esetén elég valószerűtlen látványt nyújt. A hatás olyan, mintha egy fekete koronggal letakarták volna a pupillát, ugyanis így nem tűnik domborúnak, és a szélei is túl éles határt képeznek a környezettel.

Az én megoldásomban – hogy a domború hatást megőrizsem – megtartom a pontok intenzitás értékeit, csupán színüket változtatom meg. Ezt úgy lehet elérni, hogy szürkeskálássá alakítom a pupilla pontjait. Ekkor azonban általában még elég világosszürke pupillát kapok, melyet sötétíteni kell. Annak érdekében, hogy a szem természetességét sugalló csillogása megmaradjon, a fehér, vagy ahhoz közeli pontok világosságát nem csökkentem, ennek az eljárásnak a során tulajdonképpen kontrasztemelés történik. A széleken levő pontokra érdemes környezeti átlagolást (simítást) végezni, hogy ne legyen olyan éles a határvonal.

Implementáció

Az előző fejezetekben taglalt feladatok megoldására készítettem egy szoftvert, melyben implementálásra kerültek a korrekciós algoritmusok. A továbbiakban ezen megvalósítás legfontosabb momentumait mutatom be, illetve az egyes funkciók használatának bemutatásán keresztül ismertetem a megoldások logikai menetét és az elért eredményeket.

A fejlesztői környezet

Az implementáció megvalósítása a Microsoft Visual Studio 2005 fejlesztőrendszerben történt Microsoft Visual C# programozási nyelven. A használt keretrendszer a .Net Framework 3.0, melynek prezentációs alrendszerével, a Windows Presentation Foundation (WPF) segítségével történik a megjelenítés. A képek feldolgozására a korábbi megjelenítésért felelős GDI+ API-t használom, melynek funkcionalitásának eléréséhez a System.Drawing, illetve a System.Drawing.Imaging névterek állnak rendelkezésre.

A felhasznált képfeldolgozási eszközök nagy részét az AForge.Net [15] nyílt forráskódú C# keretrendszer biztosítja, mely olyan alapvető eszközöket szolgáltat, mint az éldetektálás vagy a Hough kör transzformáció. Az arcdetektálás a C/C++ nyelven írt szintén nyílt forráskódú OpenCV [16] segítségével történik.

EasyPhotoRetouch

A korábban vázolt algoritmusok az EasyPhotoRetouch nevű saját fejlesztésű programban kerültek megvalósításra. A program a retusáló funkciókon kívül egyéb segédeszközöket is tartalmaz. Ezek közé tartoznak a fileműveletek, melyek segítségével a tárolt képek megnyithatók és a módosítások elmenthetők.

Egy másik segédeszköz a képvágó, mellyel a felhasználó könnyen ki tudja vágni a képből a számára fontos részt. További segítséget nyújt, hogy a vágóablak oldalárányait megadhatja, illetve rögzítheti, annak érdekében, hogy a kép a későbbi felhasználása (pl. nyomtatása) során a célnak megfelelő legyen.

A harmadik segédeszköz a nagyítás, mellyel a felhasználó kinagyíthatja a képének részleteit, ezáltal pontosabb kijelöléseket végezhet az egyes lépések végrehajtása során.

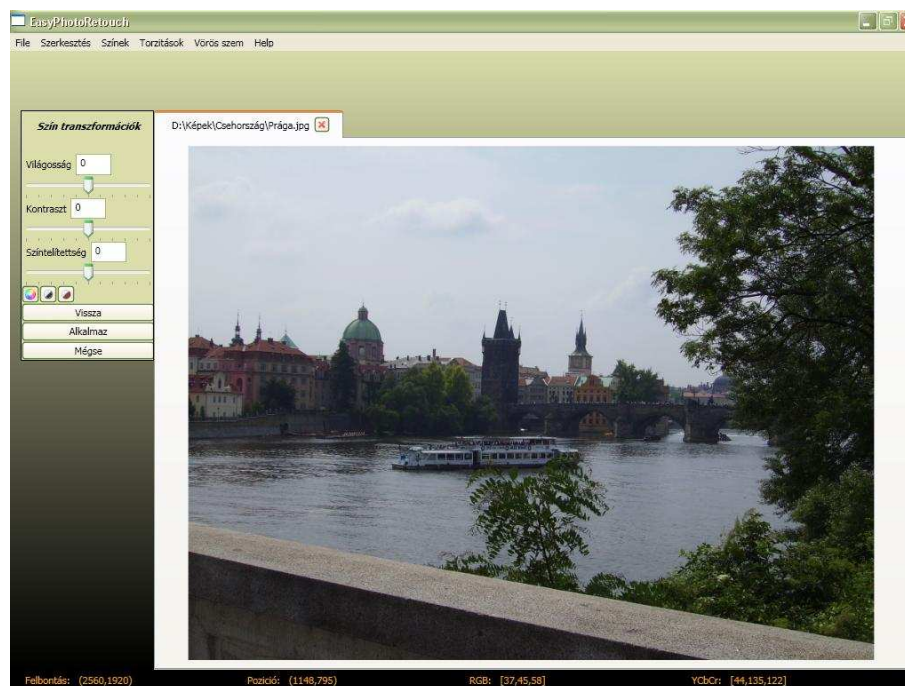
Képek betöltése, mentése

A képek betöltése valamely háttértárolóról a File menü Megnyitás pontjában lehetséges. Megjelenik a Windows szokásos file megnyitás dialógusa, ahonnan kiválasztható valamely, a program által támogatott formátumoknak megfelelő képfile. A támogatott formátumok a következők: 24 bites .bmp, .jpg, .gif, .png kiterjesztésű fileok. Megnyitás után a kép megjelenik a képernyőn, hogy azután különböző manipulációk legyenek végezhetők rajta. Egy időben több kép is lehet megnyitva, a fűlek segítségével válthatunk egyikről a másikra.

A módosítások által létrejövő képek, az előbb felsorolt formátumok bármelyikében elmenthetők a File menü Mentés pontjában. Az egyes képek bezárásakor vagy kilépéskor a program automatikusan rákérdez a még nem mentett módosítások mentésére.

Szín transzformációk

A szín transzformációk a Színek menüpontból érhető el. Ekkor baloldalon megjelenik egy panel rajta az egyes paraméterek változtatásához szükséges csúszkákkal és néhány gombbal.



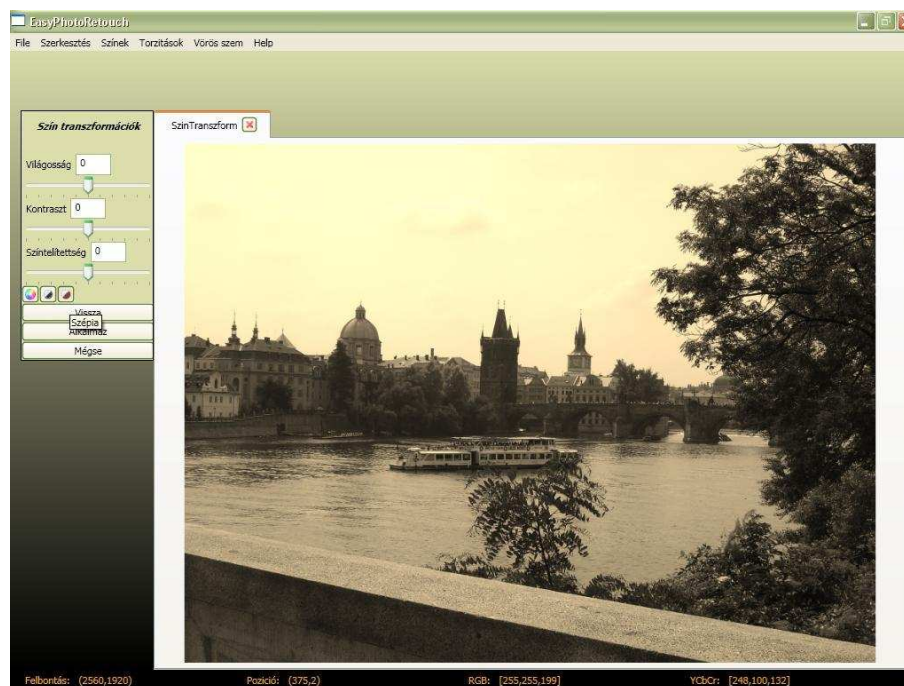
25.ábra

Az első csúszka a világosságot szabályozza, ha jobbra húzzuk a kép világosabb, ha balra, a kép sötétebb lesz. Az alatta levő csúszka a kontrasztot állítja, jobbra emeli, balra csökkenti a kép kontrasztját. Az utolsó csúszka a színtelítettség beállítását teszi lehetővé, ha balra húzzuk, csökken a színtelítettség, a legkisebb érték esetén szürkeskálás képet kapunk. A csúszka jobbra húzásával növekszik a színtelítettség.

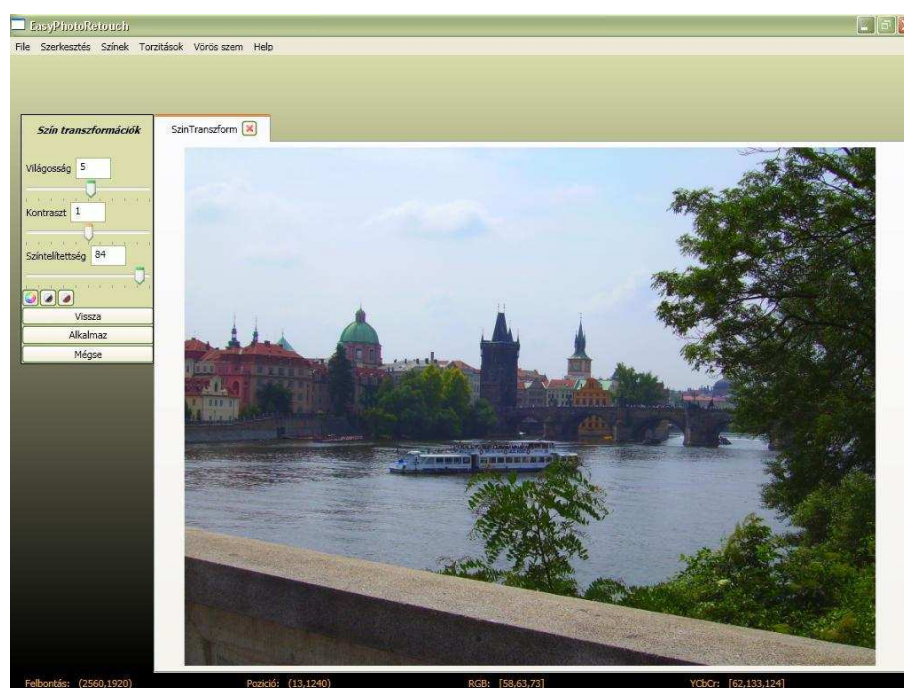
A csúszkák alatt látható három gomb, melyek sorban a következők: színes kép, szürkeskálás kép, szépia. Alapértelmezett a színes kép, ha bekapcsoljuk a valamelyik másikat, akkor annak megfelelően szürkeskálásra, vagy szépiára vált a kép.

A csúszkák mozgatása során nem történik más, minthogy az értékeknek megfelelően előállnak a ColorMatrix-ok, majd azok szorzatát alkalmazom a képre. Mivel a beállításokkal egy időben már meg is jelenítem az eredményt, a gyorsabb számítás érdekében a képet kisebb felbontásban kezelem.

Az utolsó három gomb a vezérlésért felelős. A Vissza gomb segítségével a beállítások visszaállnak az eredeti értékekre. Az Alkalmaz gomb hatására a beállításoknak megfelelően a változtatás megtörténik (az eredeti felbontásban) és létre jön az eredménykép. A Mégse gomb megnyomásával kilépünk a szín transzformációs beállításokból.



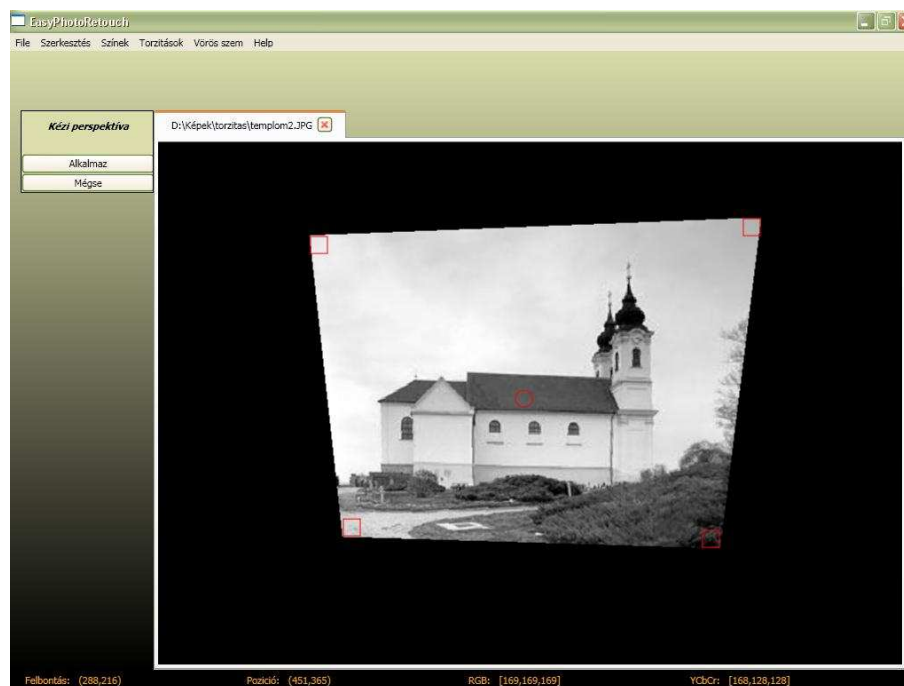
26.ábra: szépia



27.ábra: az eredeti kép javítása

Perspektívorzítás

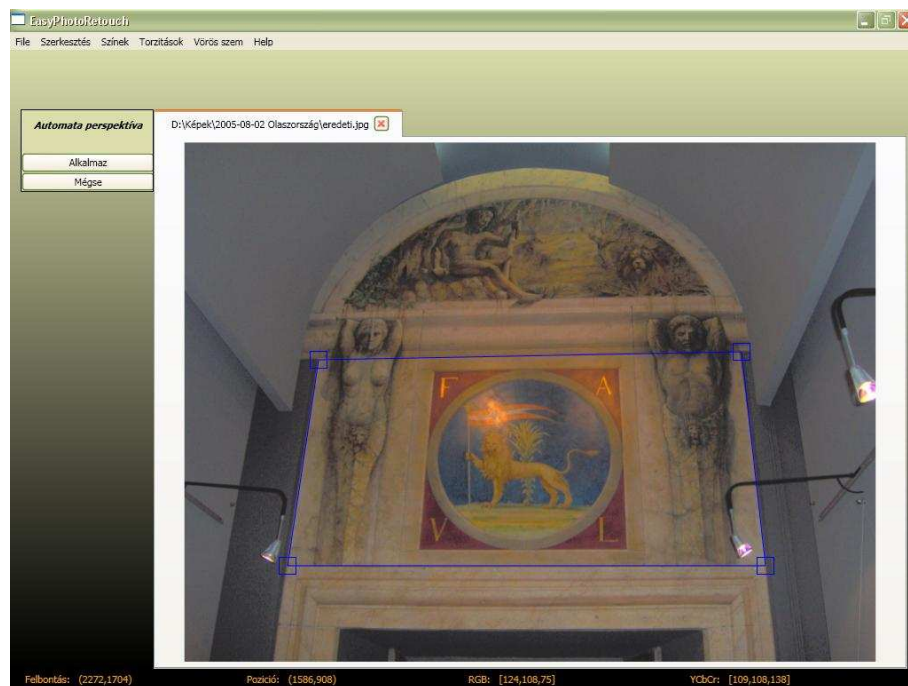
A Torzítások menüben található a Kézi perspektíva és az Automatikus perspektíva menüpontok. A kézi perspektíva kiválasztása után megjelenik a kép sarkaiban egy-egy négyszög, melyeket a felhasználó megfoghat, és szabadon húzhat, ennek megfelelően fog változni a kép alakja. Itt az történik, hogy mikor meghúzza valamely sarkot, akkor kiszámításra kerül a H mátrix, mellyel a kép eredeti téglalapját az aktuális négyszögre lehet leképezni. A leképezés azonnal meg is történik, ennek köszönhetően a felhasználó folyamatosan látja módosításának eredményét. Természetesen nagy képek esetén itt is alacsonyabb felbontáson történik a transzformáció a megfelelő sebesség elérésére. Miután sikerült megfelelően beállítani a képet, a bal oldali Alkalmaz gomb megnyomásával a transzformáció kiszámításra kerül az eredeti felbontás szerint, és létre jön az eredménykép.



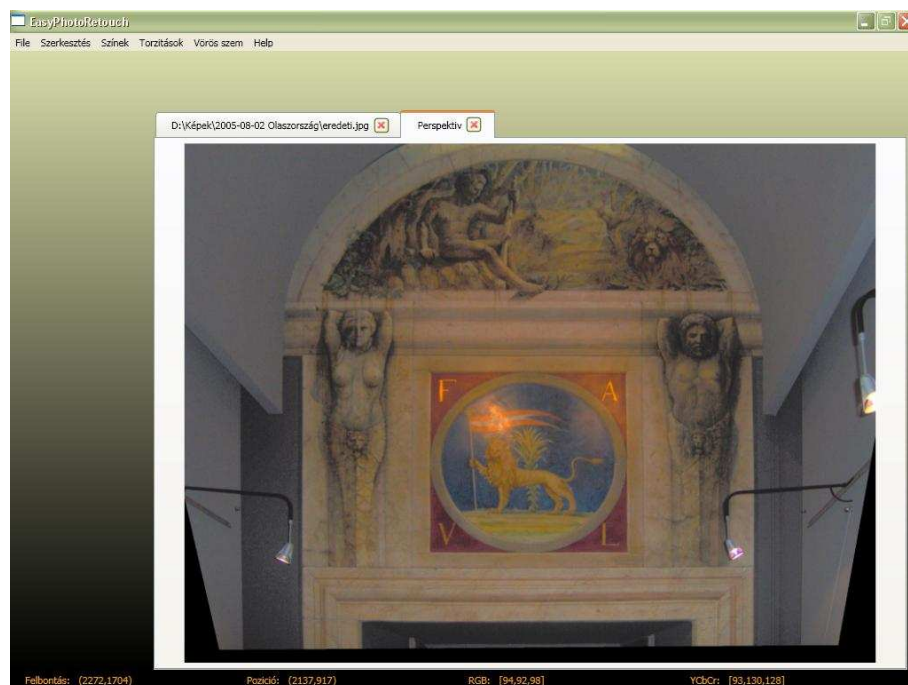
28.ábra: kézi perspektíva beállítás

Az Automata perspektíva menüpont választásával, a képen megjelenik egy téglalap, melynek sarkai megfoghatók, és húzhatók. A felhasználónak ezeket a sarkokat kell a kép megfelelő pontjaira mozgatni, ezzel kijelölve egy, a valóságban téglalap, a képen tetszőleges négyszög alakú tartományt. A kijelölés pontosítására itt érdemes használni a nagyítást, mely az egér görgőjével érhető el. Miután kijelölte a megfelelő négyszöget, és megnyomta az

Alkalmaz gombot, a program meghatározza a négyszöghöz tartozó téglalapot, kiszámítja a H mátrixot és elvégzi a transzformációt.



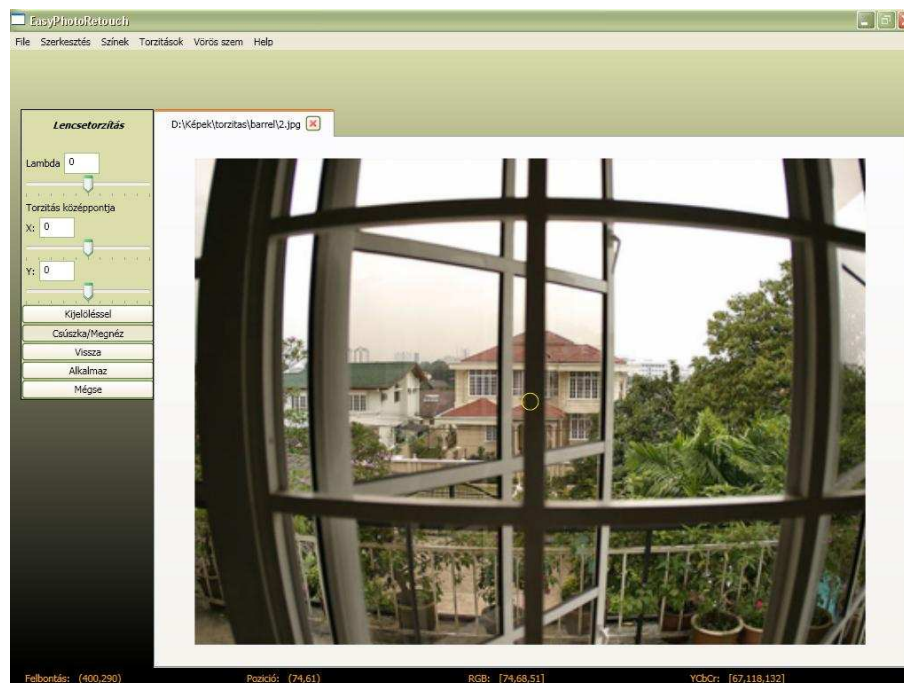
29.ábra: a sarkok elhelyezése



30.ábra: az eredményül kapott kép

Lencsetorzítások

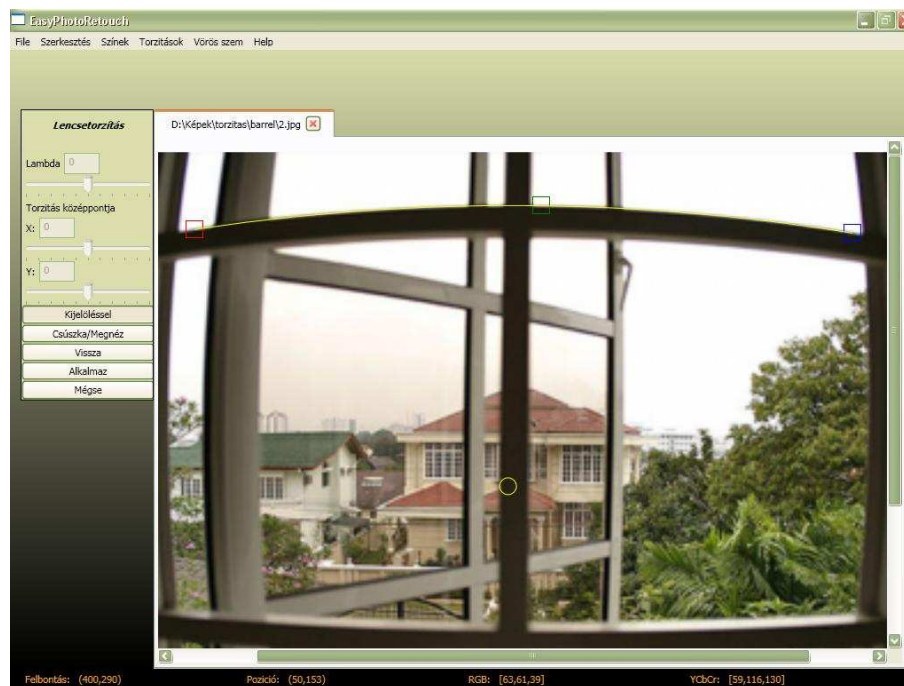
Szintén a Torzítások menüben található meg a Lencsetorzítások menüpont.



31.ábra: lencsetorzítás korrekciója

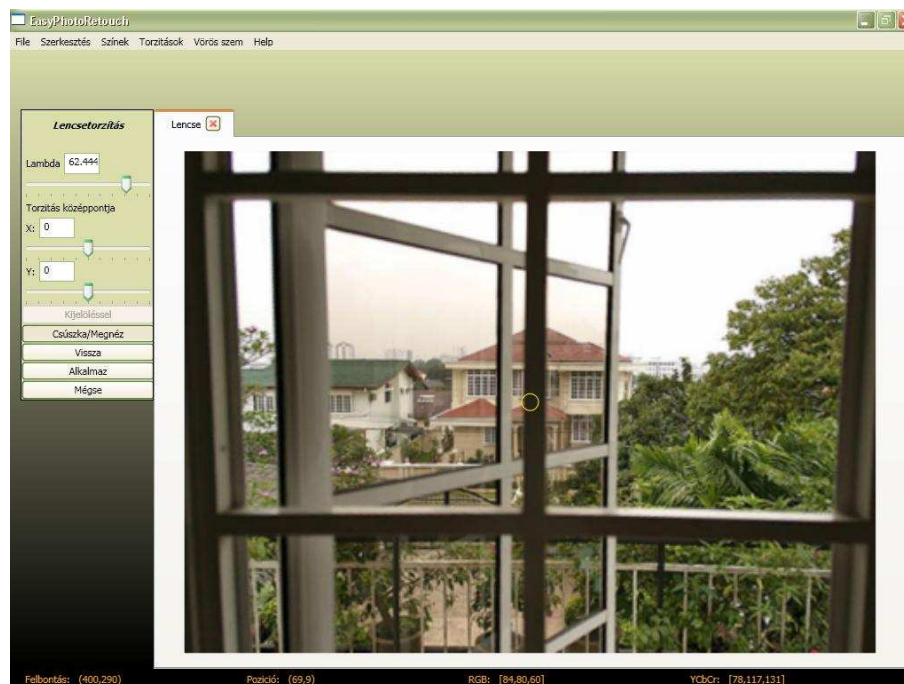
Az első csúszkával a λ paraméter értékét tudjuk explicit módon megadni, a csúszka húzásával a paraméternek megfelelő torzítás fog megjelenni a képen. Ha balra húzzuk hordó, ha jobbra, akkor párna alakú torzítás jön létre a képen. Az alatta levő két csúszkával a torzítás középpontja változtatható, melyet a képen a sárga kör jelöl. A kör megfogásával egérrel is a tetszőleges pontba húzható a torzítási középpont. Ha λ meg van adva, akkor a középpont mozgatásával a torzítás változása látható a képen.

A torzítás megadására másik mód is van. Ekkor a bal oldali gombok közül a Kijelöléssel feliratút kell megnyomni. Ezután a képre, helyezhető a torzítást kijelölő három pont.



32.ábra: a torzult egyenes megadása

A piros, zöld és kék jelölők lerakása után megjelenik az őket összekötő sárga görbe, mely segít a pontok precíz pozicionálásában. Ha a lerakott jelölők elhelyezkedése nem pontos, akkor egérrel a megfelelő helyre húzhatók, a pozíciójukhoz folyamatosan alkalmazkodik a sárga görbe. Miután kijelöltük a görbét, meg kell nyomni a Csúszka/Megnéz gombot, aminek hatására a görbe alapján meghatározásra kerül λ , ez be is íródik a felső szövegdobozba, továbbá a képen láthatóvá válik a torzítás hatása.



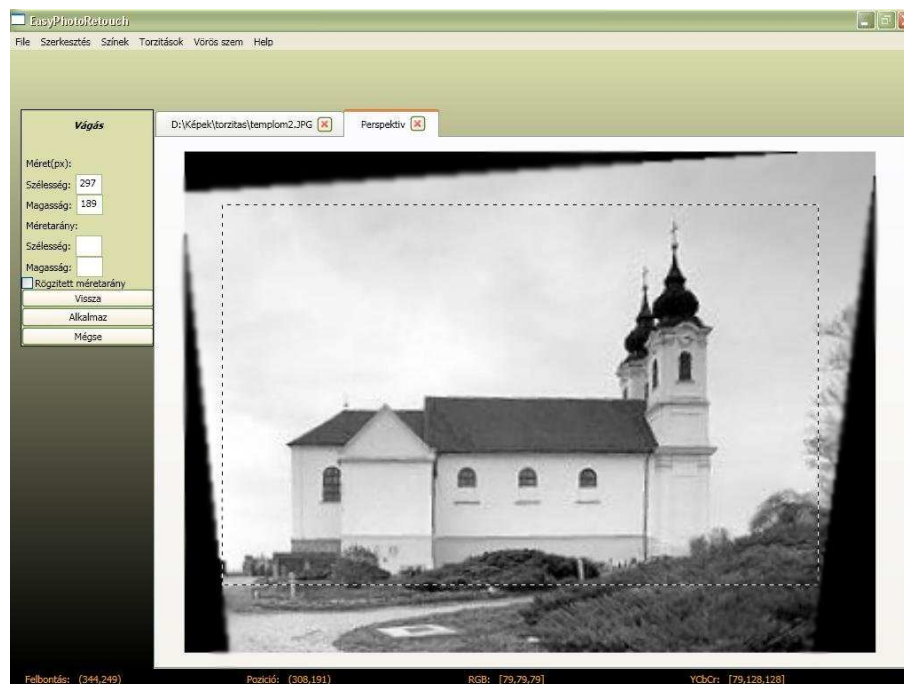
33.ábra: a korrekció eredménye

Ekkor ismét engedélyezett a csúszkák állítása, mellyel tovább finomítható az eredmény.

Ha már elégedettek vagyunk a látottakkal, akkor az Alkalmaz gomb megnyomásával az eredeti felbontású képre alkalmazza a változtatást.

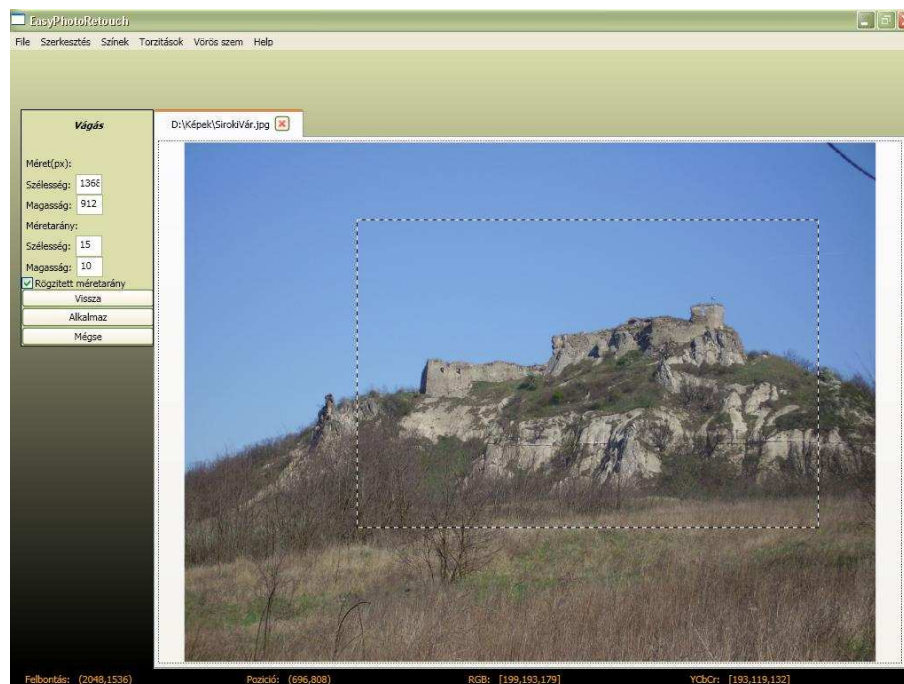
Vágás

A képek vágása a Szerkesztés/Kivágás menüpontban érhető el. A művelet több szempontból is fontos. A perspektívorzítás és a lencsetorzítások korrekciója után az eredményképen nem odatartozó feketével jelölt részek jelenhetnek meg. Mindkét esetben azért, mert az eredményül kapott kép alakja nem téglalap lenne, hanem előbbinél tetszőleges négyszög, utóbbinál egy hordó vagy párna alakú alakzat. Egy képet viszont csak téglalapban lehet ábrázolni, ezért a hiányzó részeket ki kell tölteni valamivel, ez esetben feketével. A kép vágásával kinyerhető a “hasznos rész”.



34.ábra: vágás a torzítás után

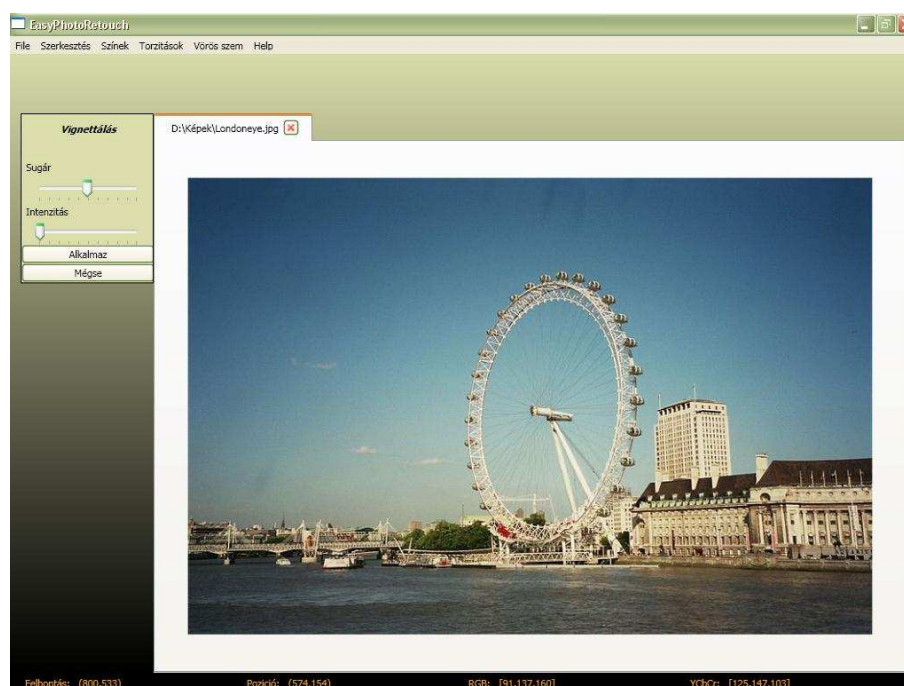
Másik jelentősége a vágásnak az, hogy ha elő akarjuk hívatni képeinket, akkor azt vagy automatikusan megvágják, vagy kiegészítik fehér szélekkel, ugyanis a digitális gépekkel készült képek oldalaránya nem egyezik meg a papírformátumú képek oldalarányával. Ekkor legjobb, ha mi vágjuk meg a képet a nekünk tetsző módon, a megfelelő arányban. Ehhez nyújt segítséget a vágóablak oldalarányának beállíthatósága, mellyel az arányokat megadhatjuk, de az ablak méretét és pozícióját szabadon változtathatjuk. A méretarány alatti mezőkbe beírjuk a szélességet és a magasságot, majd kipipáljuk a rögzített méretarány jelölőjét, ezután egérrel tetszőleges pozícióba húzhatjuk a vágóablakot, illetve a méretét is változtathatjuk, az arányok változatlanok maradnak.



35.ábra: 15x10 cm-es fényképnek megfelelő vágás

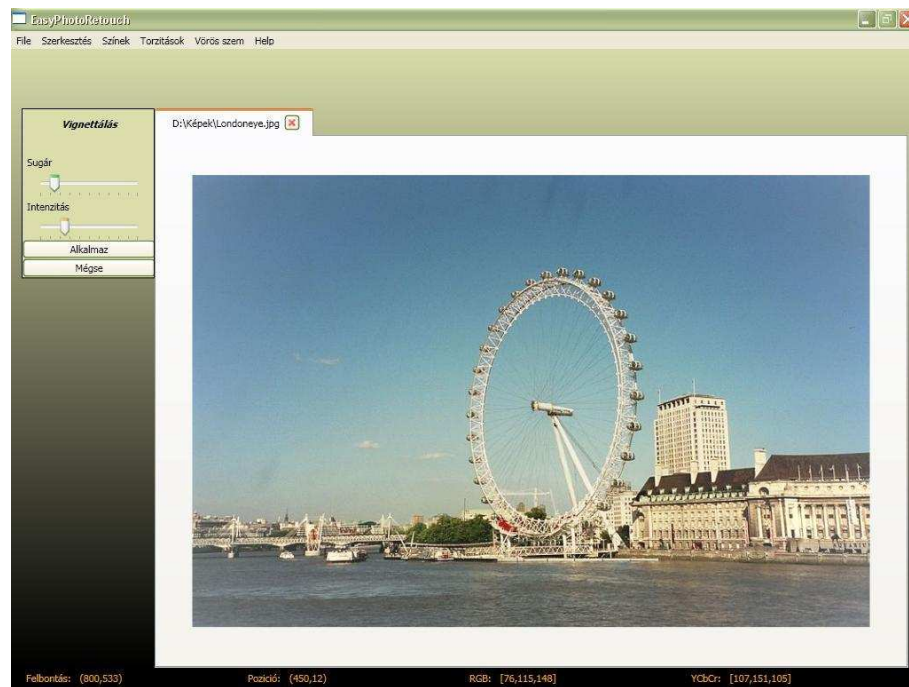
Vignettálás

A vignettálás korrekciója a Torzítások menüben található.



36.ábra: a kép besötétedik a sarkai felé

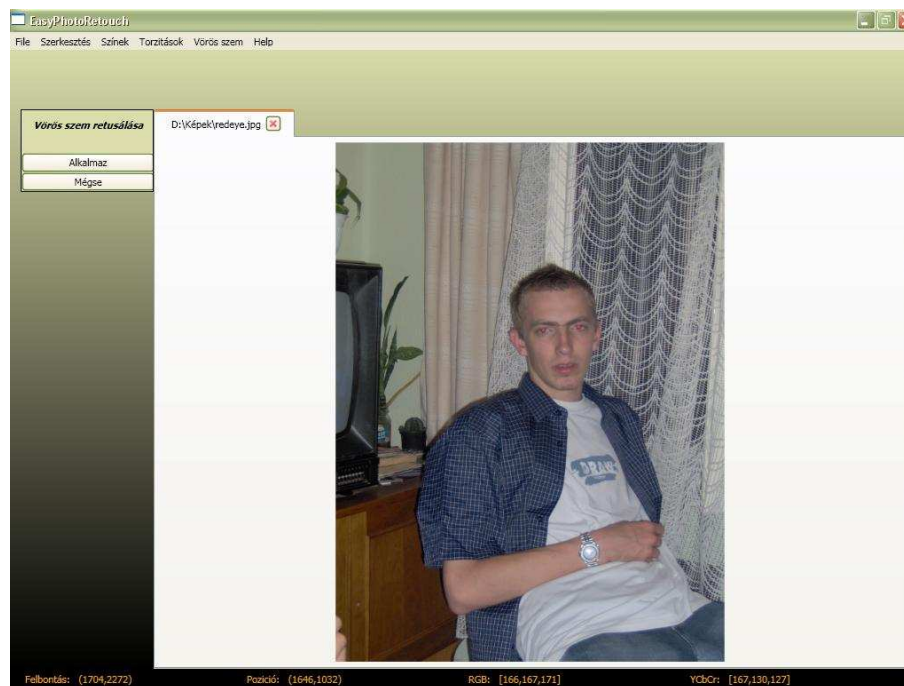
A baloldali panelen található két csúszka, a felsővel azt állíthatjuk be, hogy a kép középpontjától milyen távolságra kezdődik a vignettálás, az alsóval pedig az intenzitás-növelés nagyságát adhatjuk meg. Az értékek változtatásával az eredmény azonnal látható, így megkereshetők a megfelelő beállítások, majd az Alkalmaz gomb megnyomásával létrejön a korrigált kép.



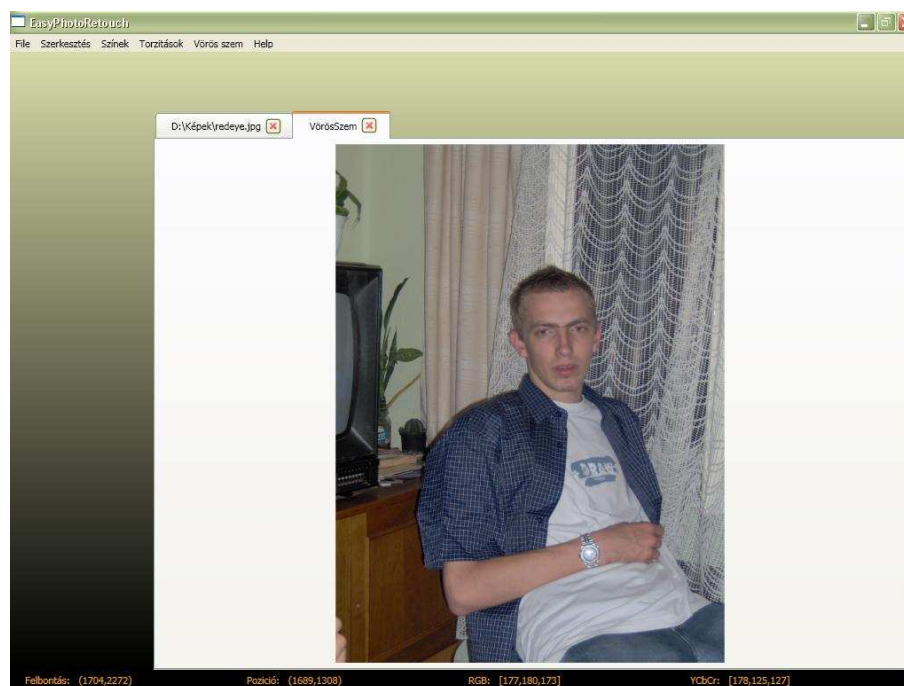
37.ábra: a megfelelő paraméterek beállítása után

Vörös szem

A vörös szem effektus megszüntetésének módja a felhasználó szempontjából meglehetősen egyszerű, mivel az algoritmus teljesen automatikus. Miután betöltöttük a vörös szemet tartalmazó képet válasszuk a Vörös szem/Vörös szem eltávolítása menüpontot, majd kattintsunk az Alkalmaz gombra.



38.ábra: *korrekció előtt*



39.ábra: *korrekció után*

Összefoglalás

Szakdolgozatom célkitűzése az egyszerű, kompakt digitális fényképezőgépek által készített képek utólagos korrekciója. Ezen fényképezők gyakori hibái a színproblémák, különböző torzítások, elmosódás, képi zajok. Az általam kiválasztott problémák a következők voltak: színekorrekciók, perspektíva torzítás korrigálása, lencsetorzítások, vignettálás, vörös szem effektus megszüntetése. A problémák bemutatása, megértése után megoldásukra kidolgoztam egy vagy esetenként több eljárást, melyek alapján egy implementáció is készült.

A színekorrekciók esetén a kép világosságának, kontrasztjának, színtelítettségének változtathatóságára manuális megoldást állítottam elő, mellyel a felhasználók tetszőlegesen beállíthatják ezeket az értékeket. További cél lehet ezen a téren az automatizálás, mely lehetővé teszi a tulajdonságok emberi beavatkozás nélküli beállítását.

A perspektíva torzítás esetén két megoldást is készítettem, az egyik teljes mértékben a felhasználó beavatkozására támaszkodik, a másik egy félautomata eljárás, melyben a felhasználónak ki kell jelölnie egy torzult téglalapot, amit azután az eljárás automatikusan egy nem torzult téglalapra képez le. A módszer teljes automatizálására speciális esetekben látok lehetőséget, például rögzített kamera esetén. Ennek alkalmazása az autópályákon a rendszámtáblákat figyelő és felismerő rendszerek segítségére lehet.

A lencsetorzítások korrekciójának elvégzésére az előzőhöz hasonlóan szintén két megoldási lehetőséget fejlesztettem ki. További fejlesztési lehetőséget a görbék pontos, de egyszerűbb kijelölése jelenthet, mely tovább növelné a megbízhatóságot és a gyorsaságot. Egy újabb megoldást jelentene az egyes fényképezőgépek torzítási mértékének összegyűjtése, ami alapján a képek felhasználói beavatkozás nélkül automatikusan korrigálhatók.

A vignettálás korrigálására szintén egy manuálisan paraméterezhető módszert készítettem. Esetenként azonban a megadható paraméterekkel nem lehet elérni maximálisan a kívánt eredményt, ennek oka az intenzitás változását közelítő függvény pontatlansága, további kutatással egy jobb közelítés elérése lehet a cél.

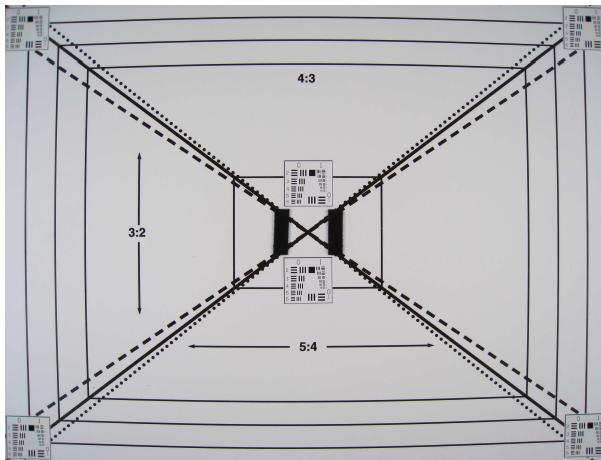
Utolsó téma a vörös szem effektus megszüntetése volt, mely a legösszetettebb és legtöbb irányból megközelíthető probléma. A megoldásomban az OpenCV arcdetektálójával megkeresem az arcokat, majd azokon kördetektálással és szín információk segítségével a vörös szemeket. Jelen eljárás csak szemből nézeti, normál helyzetű arcokra van kielezve, mely néhány plusz lépés közbeiktatásával a továbbiakban általánosítható.

Irodalomjegyzék

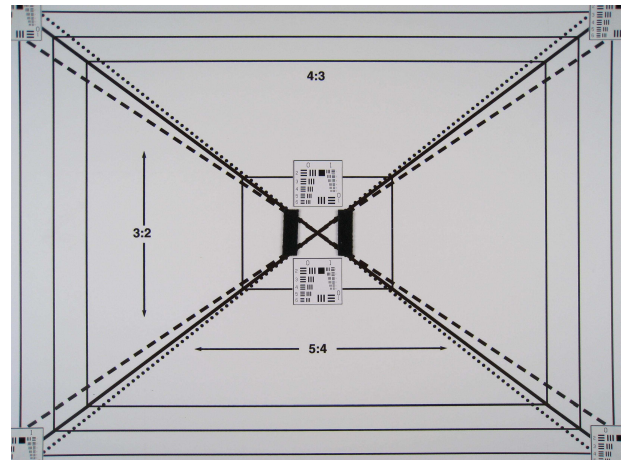
- [1] Budó – Mátrai: *Kísérleti fizika III.* Nemzeti Tankönyvkiadó
- [2] <http://en.wikipedia.org/wiki/Brightness>
- [3] http://en.wikipedia.org/wiki/HSV_color_space
- [4] [http://en.wikipedia.org/wiki/Contrast_\(formula\)](http://en.wikipedia.org/wiki/Contrast_(formula))
- [5] Strand – Hayman: *Correcting Radial Distortion by Circle Fitting*. Royal Institute of Technology (KTH), SE-100 44 Stockholm, Sweden
- [6] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/EPsrc_SSAZ/node7.html
- [7] <http://en.wikipedia.org/wiki/Supersampling>
- [8] <http://www.graficaobscura.com/interp/index.html>
- [9] [http://msdn2.microsoft.com/enus/library/system.drawing.imaging.colormatrix\(VS.71\).aspx](http://msdn2.microsoft.com/enus/library/system.drawing.imaging.colormatrix(VS.71).aspx)
- [10] Vezhnevets – Sazonov – Andreeva: *A Survey on Pixel-Based Skin Color Detection Techniques*. Moscow State University, Moscow, Russia.
- [11] Viola – Jones: *Robust Real-Time Face Detection*. International Journal of Computer Vision 57(2), 137–154, 2004.
- [12] http://www.intel.com/technology/itj/2005/volume09issue02/art03_learning_vision/p04_face_detection.htm
- [13] Freund – Schapire: *A Short Introduction to Boosting*. Journal of Japanese Society for Artificial Intelligence.
- [14] Gaubatz – Ulichney: *Automatic Red-eye Detection and Correction*. Cornell University, Hewlett-Packard Co.
- [15] <http://code.google.com/p/aforge/>
- [16] <http://www.intel.com/technology/computing/opencv/>

Függelék

További tesztképek és eredmények



a)



b)

a) Canon Powershot fényképező kalibrációs képe, látható hordótorzítással; b) a torzítás korrigálva



a)



b)

a) hordótorzítás Konica Minolta géppel; b) korrigálva, de perspektívtorzítás látható



Perspektívtorzítás is korrigálva, színkorrekciók és vágás



a)



b)

a) hordó alakú torzulás; b) korrigálva



a)

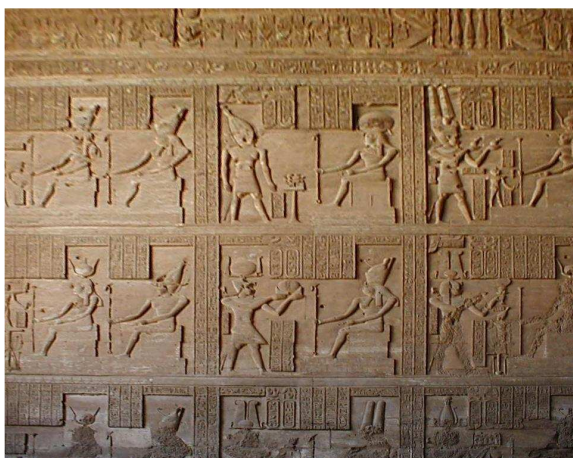


b)

a) perspektívtorzítás; b) korrigálva



a)



b)

a) perspektívtorzítás; b) korrigálva

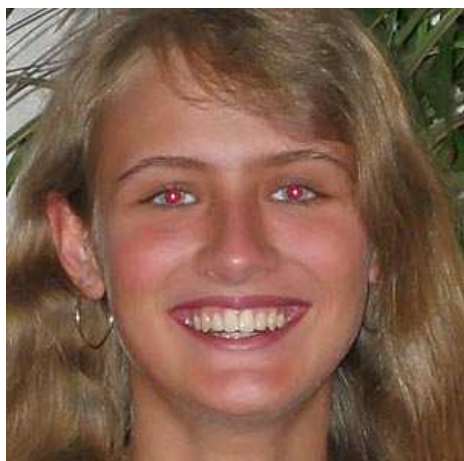


a)



b)

a) vignettálás; b) korrigálva



a)



b)

a) vörös szem effektus; b) javítva



a)

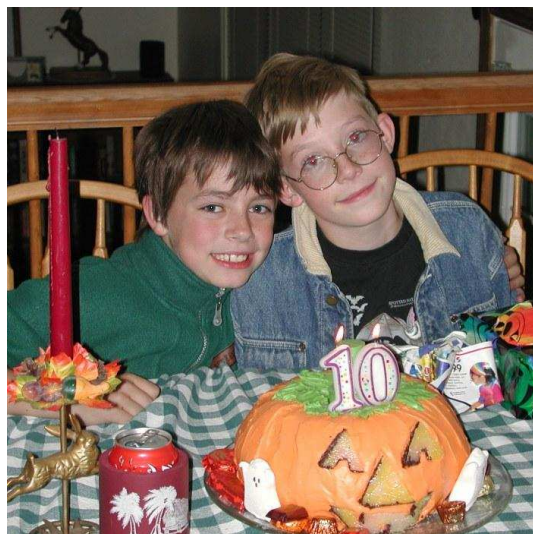


b)

a) több arc vörös szemekkel; b) javítva

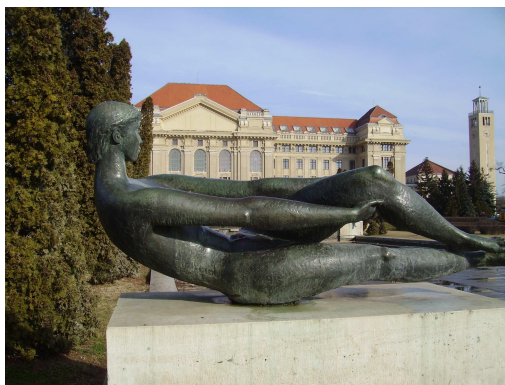


a)



b)

a) a jobb oldali gyerek arca oldalra billen; b) az ő arcát nem találja meg a detektor



a)



b)

a) szobor; b) színkorrekció és vágás után